



IIC2513 – Tecnologías y Aplicaciones Web

Examen

Instrucciones: Sea preciso: no es necesario escribir extensamente pero sí ser preciso. En caso de ambigüedad, utilice su criterio y explicita los supuestos que considere convenientes. Esta interrogación fue diseñada para durar 120 minutos. Escriba su nombre en cada una de las hojas que entregue.

1. (0.8 pts) Considere el siguiente *end-point* de una *API* web para obtener el detalle de un artista:

Request:

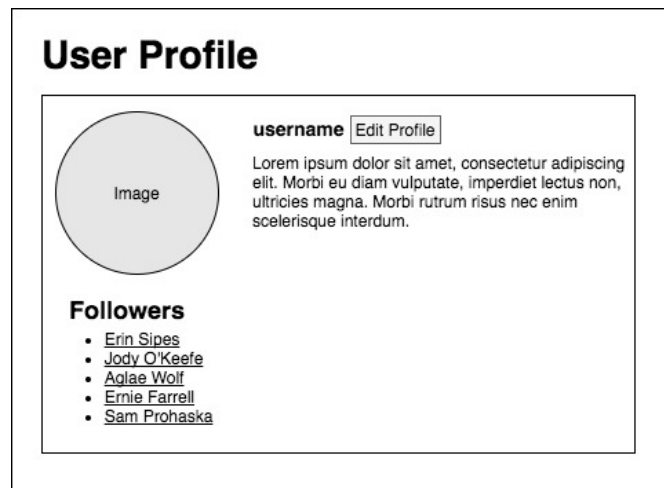
POST `http://music.com/api/3.0/artists/{artist_id}/get_detail.json?apikey={your_api_key}`

Response:

```
{
  "status": "ok",
  "data": {
    "name": "Band of Horses",
    "genres": ["indie folk", "indie pop"],
    "popularity": 59,
    "id": "00dUWJ0sBjDrqHygGUXeCF"
  }
}
```

En base a la petición y la respuesta de este servicio web, explique qué características *RESTful* no se están respetando.

2. (0.4 pts) Explique 2 ventajas y 2 desventajas/desafíos que presenta una *single-page application*.
3. (0.6 pts) Escoja 2 tipos de ataques de la siguiente lista y explique en qué consisten:
- *Session Hijacking*
 - *Session Fixation*
 - *Cross-Site Request Forgery (CSRF)*
 - *Cross-Site Scripting (XSS)*
 - *Phishing*
 - *CSS Injection*
4. (1.0 pts) Considere la siguiente imagen:



Se le pide que implemente el *HTML* y *CSS* necesarios para lograr ese resultado.

Notas:

- Los nombres de los *followers* son enlaces.
- No considere el borde externo de la imagen, no es parte de la página web.
- Puede elegir entre utilizar *HTML* o *ERB* y *CSS* o *SCSS*.
- No necesita escribir todo el contenido que se ve en la imagen en su *HTML*. Concéntrese en la estructura y utilice *placeholders*.
- Asigne tamaños y márgenes proporcionales a la representación de los elementos como se ven en la imagen.

5. (1.0 pts) Considere una aplicación *Rails* que cuenta con un *scaffold* de *users*, cuyos atributos son *email* y *password*. Además, el modelo *User* implementa un método *authenticate* que permite validar las credenciales de un usuario.

Sin utilizar gemas adicionales, detalle los cambios que se deben realizar en la aplicación para permitir que estos usuarios puedan iniciar y cerrar sesión a través de un navegador. En su respuesta no se pide que implemente código, pero sí que especifique los nombres y responsabilidades de los archivos, clases y métodos que se requieran.

6. (0.4 pts) Considere la siguiente estructura *HTML*:

```
<div id="main-container">
  <div class="element">
    A
  </div>
  <div class="element">
    B
  </div>
  <div class="element">
    C
  </div>
</div>
<button id="select-all">Seleccionar todo</button>
<button id="unselect-all">Deseleccionar todo</button>
```

Escriba un código *JavaScript* con la ayuda de *jQuery* que implemente las siguientes funcionalidades:

- al hacer *click* sobre un “div” con clase “element”, se le debe agregar la clase “active” a este elemento y quitársela a todos los demás
- al hacer *click* sobre un “div” con clase “element” que ya tenga la clase “active”, esta debe desaparecer de ese elemento
- al hacer *click* sobre el botón con id “select-all”, se debe agregar a todos los “div” con clase “element” la clase “selected”
- al hacer *click* sobre el botón con id “unselect-all”, se debe quitar la clase “selected” de todos los elementos “div” de clase “element”

NOTAS:

- Algunas funciones de *jQuery* son:
 - `.on(events,[selector],[data],handler)`
 - `.val([value])`
 - `.text([text])`
 - `.removeClass([className])`
 - `.addClass([className])`
 - `.attr(attributeName,[value])`
 - `.find(selector)`

7. (0.6 pts) Considere una aplicación *Rails*, la cual contiene los siguientes extractos de los archivos “app/views/users/_form.html.erb” y “app/controllers/users_controller.rb” respectivamente:

```
<%= form_for(user) do |f| %>
  <div class="field">
    <%= f.label :email %>
    <%= f.email_field :email, required: true %>
  </div>
  <div class="field">
    <%= f.label :password %>
    <%= f.password_field :password, required: true %>
  </div>
  <div class="field">
    <%= f.label :password_confirmation %>
    <%= f.password_field :password_confirmation, required: true %>
  </div>
  <div class="actions">
    <%= f.submit %>
  </div>
<% end %>
```

```
# POST /users
def create
  @user = User.new(user_params)

  respond_to do |format|
    if @user.save
      format.html do
        redirect_to @user, notice: 'User was successfully created.'
      end
    else
      format.html { render :new, status: 422 }
    end
  end
end
```

Suponiendo que la información enviada por el formulario se procesa en ese método del controlador, especifique todos los cambios necesarios en la aplicación para permitir el procesamiento de esta petición a través de *AJAX*. Considere que al cliente se le debe notificar si la petición fue exitosa o si hubo un error.

Notas:

- No se pide que implemente todo el código necesario, pero sí que especifique los comandos principales y los archivos involucrados.
- Puede elegir la política que estime conveniente para transferir la información entre servidor y cliente.

8. (1.2 pts) Considerando la siguiente descripción:

La empresa *ArriendoRápido* ofrece arriendos de maquinaria pesada para la minería. Debido a su gran crecimiento en el último tiempo, la empresa necesita una aplicación web para facilitar la administración de sus servicios. La idea es que solamente el personal administrativo de la empresa tenga acceso a la plataforma. A través de ella, los usuarios podrán asignar arriendos para una máquina a nombre de un cliente, en un horario específico. Además, *ArriendoRápido* ofrece los servicios de un operario experto de la máquina arrendada, que se podría incluir en un arriendo con un costo adicional. Por otra parte, la empresa exige que para cada reserva se ingrese un pago por adelantado, que corresponde a un porcentaje del costo. Este porcentaje es variable y puede dividirse en varios pagos, dependiendo del historial con el cliente. Por último, a partir de la modelación del sistema la empresa espera obtener qué máquinas y qué operarios han sido reservados por un cliente.

- a) Realice un diagrama entidad-relación que represente el modelo del problema (incluyendo los atributos necesarios que se desprenden del enunciado).
- b) Nombre los modelos de *Rails* necesarios para representar esta modelación. Además, escriba las declaraciones necesarias en cada modelo para reflejar las asociaciones a nivel de *Active Record* (solamente las líneas de código para las asociaciones en cada archivo). Considere que el requerimiento “obtener qué máquinas y qué operarios han sido reservados por un cliente” se tiene que obtener directamente de un cliente a través de *Active Record*.
- c) Especifique los nombres de las tablas y columnas en la base de datos necesarias para crear las asociaciones entre los modelos.

Nota:

- Puede nombrar los elementos en inglés o español, siempre y cuando respete las convenciones de *Rails*.
- Especifique todos los supuestos que crea necesarios.