



## IIC2513 – Tecnologías y Aplicaciones Web

### Interrogación 2

**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 100 minutos. Escriba su nombre en cada una de las hojas que entregue.

1. (0.7 pts) En el contexto del *framework Rails*, dada la definición de ‘resources :users’ en el archivo *config/routes.rb*:
  - a. Especifique las rutas que se generan, detallando método HTTP, *path* de la URL y nombre del URL *helper* correspondiente (**tip**: responda con la ayuda de una tabla).
  - b. ¿Cuál es el nombre del controlador asociado a estas rutas?
  - c. Asocie cada ruta generada a la acción correspondiente en el controlador.

a, c)

| URL Helper     | HTTP Method | URL Path        | Actions       |
|----------------|-------------|-----------------|---------------|
| users_path     | GET         | /users          | users#index   |
|                | POST        |                 | users#create  |
| new_user_path  | GET         | /users/new      | users#new     |
| edit_user_path | GET         | /users/:id/edit | users#edit    |
| user_path      | GET         | /users/:id      | users#show    |
|                | PATCH       |                 | users#update  |
|                | PUT         |                 | users#update  |
|                | DELETE      |                 | users#destroy |

b) *UserController*, *Users* o *users\_controller.rb*

2. (0.7 pts) Distinga 3 capas de validaciones que procesan los datos de una petición HTTP en una aplicación web. Para cada capa, explique cuál es el principal beneficio que aporta realizar validaciones en ese nivel.

- Cliente: evita peticiones HTTP inválidas para el servidor de aplicación, disminuyendo la carga sobre este.
- Servidor de aplicación: agrega validaciones a nivel de negocio, integrando los datos en el entorno del servicio.
- Base de datos: garantiza la integridad de los datos (esto incluye unicidad).

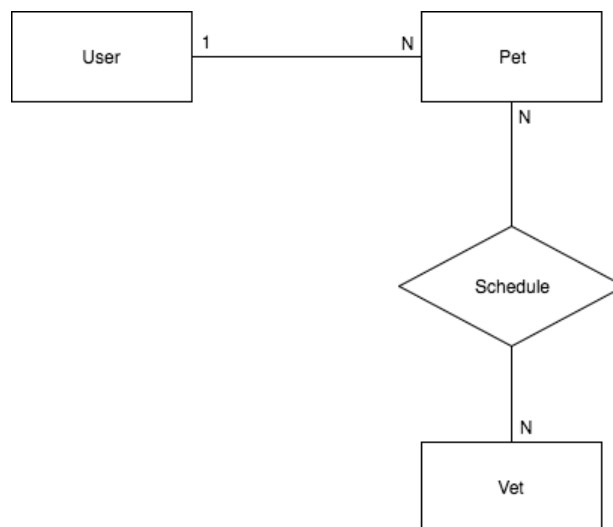
3. (1.5 pts) Considere la siguiente descripción:

Se desea desarrollar una aplicación web para que los dueños de mascotas reserven horas al veterinario. En este contexto, los usuarios pueden ingresar todas sus mascotas en la plataforma, pero por simplicidad un animal pertenece solamente a un usuario. Para cada una de sus mascotas, un usuario puede reservar horas de atención con distintos veterinarios. Cabe destacar que es importante almacenar el historial de las atenciones de los animales, para así llevar un registro en la plataforma.

En el contexto del *framework Rails*:

- Realice un diagrama entidad-relación **sin** considerar los atributos de las entidades.
- 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).
- Especifique los nombres de las tablas y columnas en la base de datos necesarias para crear las asociaciones.

a)



b)

User

- has\_many :pets

Pet

- belongs\_to :user
- has\_many :schedules
- has\_many :vets, through :schedules

Schedule

- belongs\_to :pet
- belongs\_to :vet

Vet

- has\_many :schedules
- has\_many :pets, through :schedules

c)

users

- id

pets

- id
- user\_id

schedules

- id
- pet\_id
- vet\_id

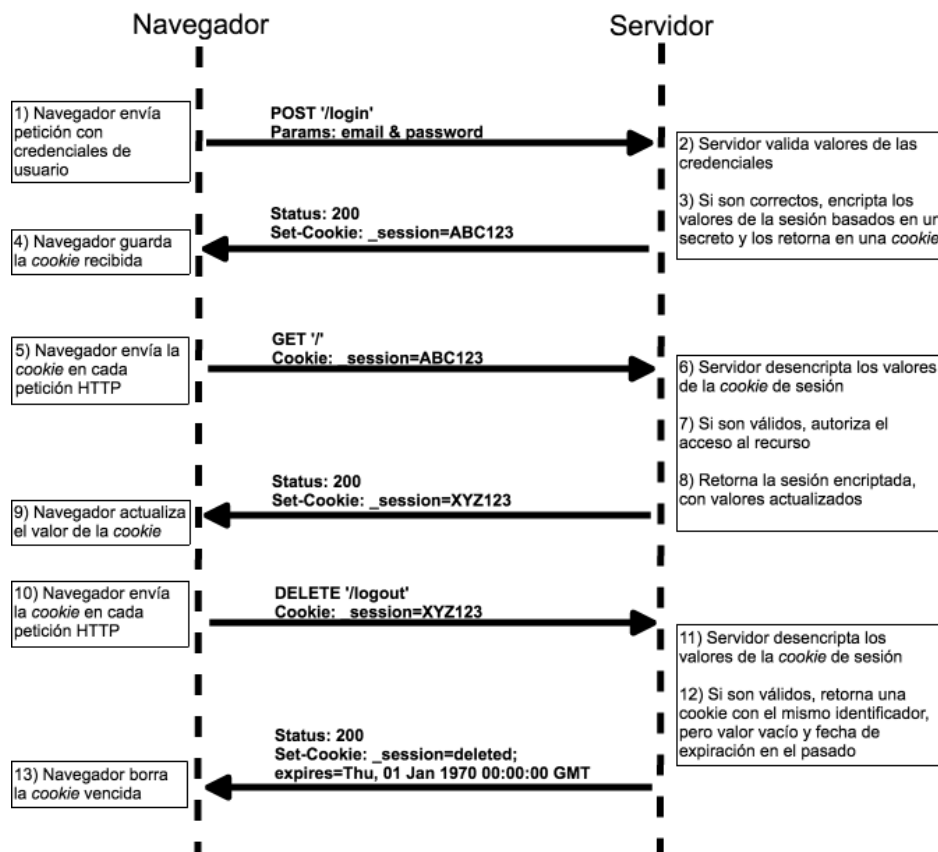
vets

- id

4. (1.3 pts) Cookies y sesiones:

- a. ¿Qué es una *cookie* HTTP?
  - b. Explique 2 estrategias para modelar el “manejo de sesión” en una aplicación web.
  - c. Describa la interacción (i.e. los pasos) entre un navegador y un servidor para mantener la sesión de un usuario, utilizando la estrategia por defecto de *Rails*. Detalle los *headers* HTTP utilizados en cada interacción.
- a) Una *cookie* HTTP es un pequeño extracto de información enviado por un servidor web para ser almacenado en un navegador de un cliente. De esta forma, un cliente puede enviar esa misma información en futuras peticiones a ese servidor.
- b) - hash de sesión: El servidor envía una *cookie* encriptada con todos los valores de la sesión del cliente. De esta forma cuando el servidor recibe una petición HTTP acompañado de esta *cookie* puede desencriptar el valor y recuperar la sesión.
- identificador de sesión: El servidor almacena en base de datos los valores de la sesión y los asocia con un identificador único. Luego, envía una *cookie* con el identificador de la sesión para que lo almacene el cliente. Cuando el servidor recibe una petición HTTP con un identificador de sesión, este busca en la base de datos los valores de la sesión del cliente a partir del identificador.

c)



5. (0.8 pts) Escriba un extracto de HTML solamente con los *tags* necesarios para que todos los siguientes selectores CSS retornen por lo menos un elemento:

- body header nav#main-menu > a
- main div#content-wrapper p.red.centered > span
- div#content-wrapper > div#users + h4

```
▼ <body>
  ▼ <header>
    ▼ <nav id="main-menu">
      <a></a>
    </nav>
  </header>
  ▼ <main>
    ▼ <div id="content-wrapper">
      ▼ <p class="red centered">
        <span></span>
      </p>
      <div id="users"></div>
      <h4></h4>
    </div>
  </main>
</body>
```

6. (1.0 pts) Escoja la alternativa correcta:

a. ¿Cuál es la sintaxis correcta para un *link*?

- <a url="www.google.com">Google</a>
- <a href="www.google.com" Google />
- <a url="www.google.com" Google />
- iv. <a href="www.google.com">Google</a>

b. ¿Qué *tag* representa el encabezado con mayor importancia?

- <h6>
- ii. <h1>
- iii. <h0>
- iv. <h7>

c. ¿Qué *tag* representa una lista enumerada?

- <list>
- ii. <ul>
- iii. <ol>
- iv. <dl>

d. ¿Qué DOCTYPE especifica que un documento utiliza HTML5?

- i. `<!DOCTYPE html>`
- ii. `<!DOCTYPE HTML5>`
- iii. `<!DOCTYPE HTML PUBLIC "http://www.w3.org/TR/html5/strict.dtd">`

e. ¿Qué elemento representa el pie de página?

- i. `<bottom>`
- ii. `<footer>`
- iii. `<header>`
- iv. `<section>`

f. ¿Qué elemento representa un menú de navegación?

- i. `<menu>`
- ii. `<nav>`
- iii. `<navigation>`
- iv. `<navigate>`

g. ¿Cómo se selecciona un elemento con *id* “*first*”?

- i. `#first`
- ii. `.first`
- iii. `/first`
- iv. `+first`

h. ¿Cómo se selecciona un elemento con clase “*warning*”?

- i. `#warning`
- ii. `.warning`
- iii. `/warning`
- iv. `+warning`

i. ¿Qué sintaxis es correcta en CSS?

- i. `{body;color:black;}`
- ii. `{body:color=black;}`
- iii. `body{color:black;}`
- iv. `body:color=black;`

j. ¿Cómo se seleccionan todos los `<p>` que tengan como ancestro un `<div>`?

- i. `div.p`
- ii. `div p`
- iii. `div + p`
- iv. `div > p`