

Apellido y Nombre:..... Legajo:.....

D'Avatar

Crear un perfil público para cada red social, aplicación o servicio en la nube que utilizamos puede ser muy tedioso: cada vez que nos registramos en algún sitio tenemos que dejar datos como nuestro email, nombre, apellido, foto de perfil, o incluso una breve descripción de nuestra persona.

Por eso diseñaremos un nuevo sistema, llamado D'Avatar, que tendrá dos grandes funcionalidades: editar y consultar un perfil público

Edición de perfil público

Estando registrado y habiendo iniciado sesión, podremos editar **nuestro** perfil, el cual se compone de:

1. Nombre y apellido
2. Una breve descripción
3. Uno o más emails, que podremos marcar como *principales* o *secundarios*: mientras que los secundarios permiten consultar al perfil usando dicho email como clave, los principales además se muestran como parte del perfil
4. Un avatar icónico que se podrá generar y editar desde D'Avatar (ver a continuación). D'Avatar **no permitirá** al usuario subir fotos como imágenes de perfil.

Para la generación de Avatares, D'Avatar contará con un editor, similar al de *faceyourmanga.com*:



Desde éste podremos generar una representación pictórica, eligiendo uno o más *estilos* para cada *atributo* de un rostro:

- los atributos soportados serán: *cabello, ojos, orejas, piel, nariz, boca*
- los estilos soportados dependerán de cada atributo. Por ejemplo el atributo *ojos* soporta como estilos *color* (seleccionable de una paleta) y *tamaño* (chicos, normales, grandes).

Además, el editor soportará agregar accesorios como corbatas, lentes y aros.

Es importante que una vez confirmada esta imagen de avatar, ésta sea fácilmente editable: si más tarde nos arrepentimos y queremos sacar un accesorio o cambiar el estilo del cabello, no deberíamos tener que volver a crear el avatar desde cero.

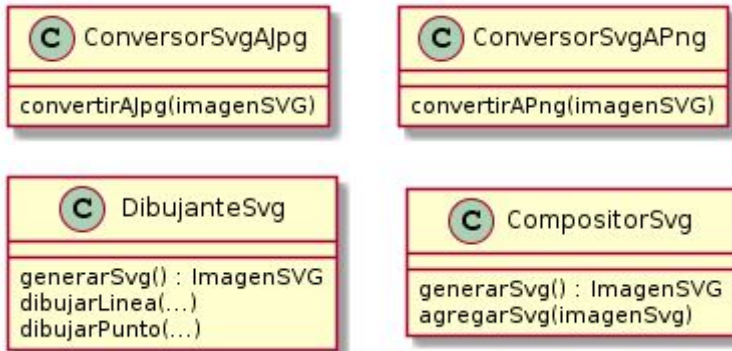
Consulta del perfil público

El perfil público se podrá consultar (sin necesidad de estar registrado) tanto a través de una interfaz gráfica web, como utilizando un API Rest, lo cual permitirá tanto a usuarios como a servicios externos consultar el perfil, buscándolo con sus mails primarios o secundarios.

En cualquiera de los dos escenarios, las imágenes se podrán obtener en cuatro formatos diferentes:

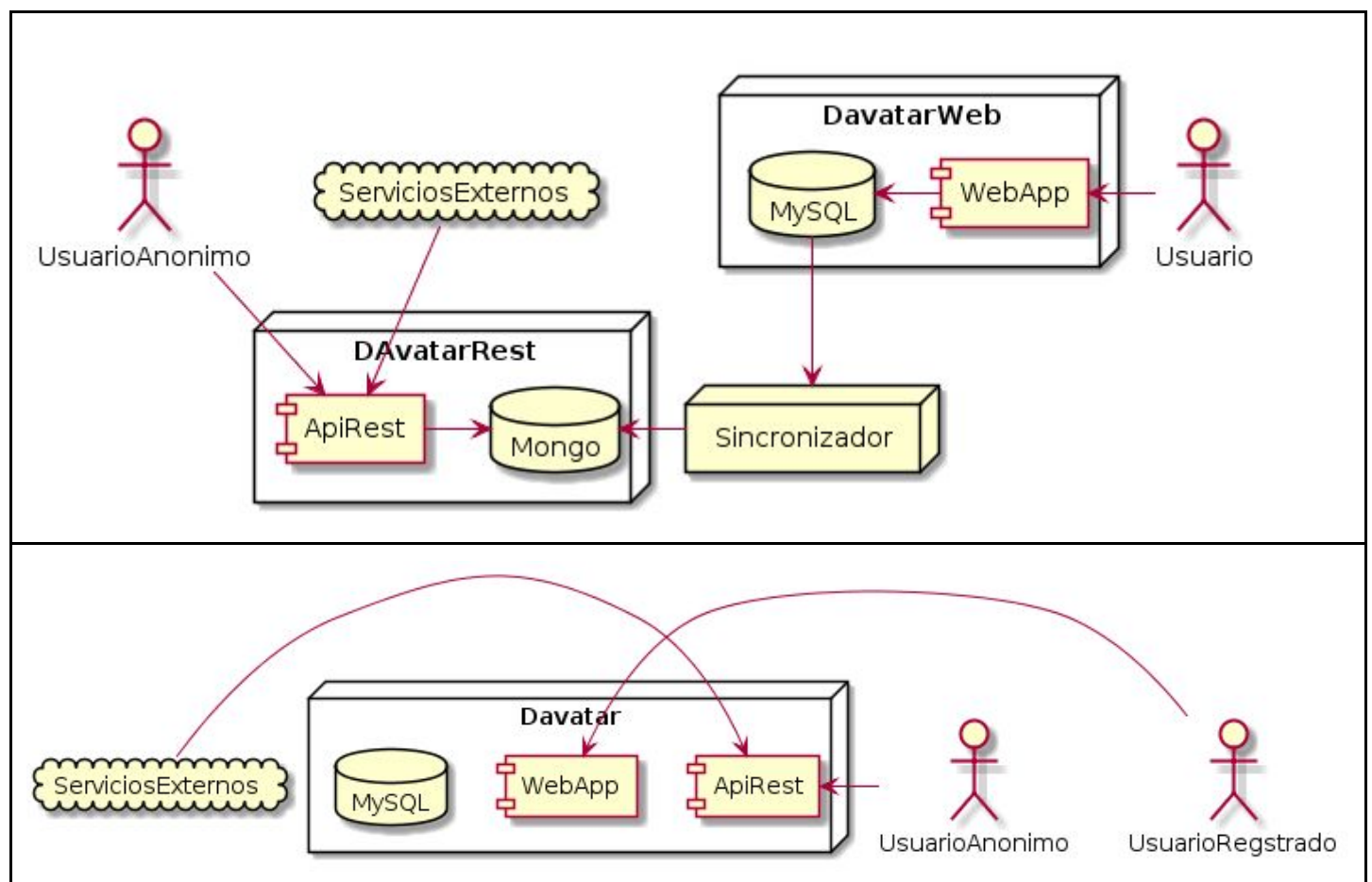
- Textual: la imagen se representará en prosa, como por ejemplo "Juan tiene ojos grandes y oscuros, tiene lentes, (...)"
- Gráfica: ya sea en formato PNG, JPG o SVG

Para lograr esto supondremos que contamos con los siguientes componentes (los cuales podemos usar todos o sólo algunos):



Aclaración: SVG es un formato de imagen *vectorial*, basado en XML, en el que cada línea, punto o forma geométrica se representa mediante un tag, lo que permite dibujar y combinar imágenes fácilmente)

Punto 1: Arquitectura (30 Puntos)



1. Compará ambas arquitecturas indicando ventajas y desventajas de cada una. ¿Cuál creés que es la más apropiada para el dominio presentado? ¿Por qué?
2. Bonus: describí la o las rutas REST que sean necesarias para el segundo caso de uso y mostrá un ejemplo de pedido y respuesta de un perfil público. ¿Cómo permitirá el API consultar las imágenes en los cuatro formatos antes mencionados?

Punto 2: Modelado en Objetos (35 Puntos)

Modelar el dominio presentado utilizando el paradigma orientado a objetos, comunicando su solución mediante un diagrama de clases (**obligatorio**), código, pseudocódigo, otros diagramas y/o prosa (complementario). No desarrollar el código ni la estructura de clases del componente visual que permite la edición de avatares, pero sí de los objetos y clases del dominio con el que este opera.

Punto 3: Persistencia (35 Puntos)

Utilizando un DER (**obligatorio**), código anotado y/o prosa (complementario), se pide explicar cómo persistir el modelo de objetos del punto anterior en una base de datos relacional, indicando claramente:

- Qué elementos del modelo es necesario persistir.
- Las claves primarias, las foráneas y las restricciones (constraints) según corresponda.
- Estrategias de mapeo de herencia utilizadas, si fueran necesarias.
- Las estructuras de datos que deban ser desnormalizadas, si corresponde.
- Justificaciones sobre las decisiones de diseño tomadas anteriormente.

Explicar supuestos y justificar decisiones de diseño.

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección: (A) otorga 30 puntos, (B) 35 puntos y (C) 35 puntos.