



10 Pines

Diseño a la Gorra - Episodio 12

Buenos Aires vs. (London vs. Chicago) - Parte 4B



Hernán Wilkinson



hernan.wilkinson@10pines.com



@HernanWilkinson

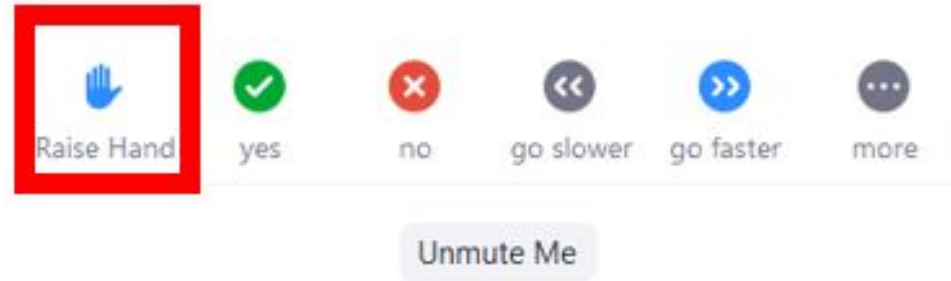


<https://alagorra.10pines.com>





Estar muteados a menos que sea necesario



No voy a poder leer el chat



La comunicación visual es importante. Usarla a discreción





Diseño ¡a la gorra!

¡Bienvenidos!

Durante esta serie de Webinars exploraremos *qué* significa **Diseñar Software con Objetos** y *cómo* lo podemos hacer cada vez mejor.

Te esperamos todos los Martes a las 19 Hrs GMT-3 a partir del Martes 11 de Agosto de 2020. Para poder participar tenes que **REGISTRARTE ACA.**

Trataremos muchos temas que irán desde cuestiones filosóficas como qué significa Diseñar en nuestra profesión y dónde está expresado ese Diseño, pasando por consejos y heurísticas para diseñar "mejor" para terminar con ejemplos concretos de cómo aplicar esas heurísticas en la *vida real*.

Los webinars son "*language agnostic*", o sea que no dependen de un lenguaje de programación en particular, aunque los ejemplos que usaremos estarán hechos principalmente en **Java**, **JavaScript**, **Ruby**, **Python** y mi querido **Smalltalk** cuando amerite 😊.

Todo el código y presentaciones estarán disponibles para que lo puedan usar y consultar en cualquier momento [acá](#).

¡Trae ganas de aprender y pasarla bien!

Donaciones



\$250 - Una buena 🍺

Pagar

\$500 - Menos que 🍔 + 🍕

Pagar

\$1.000 - Dos buenos 🍷

Pagar



Donate



¿Querés donar un monto variado o en otra plataforma?

<https://alagorra.10pines.com>

Buenos Aires vs. (London vs. Chicago)

Parte 4B - Impacto de cambios



Dónde nos quedamos



Algunos refactorings

- Refactorings:
 - Usar Json en vez de String como input del RestReceptionist
 - Usar ModelException en vez de RuntimeException
 - Creación de UserCard para sacar diccionarios
 - Mejora en el uso de Optional → Facu



	London	Chicago	Bs As
Usuario con userName vacio	Lo Permite	Lo Permite	400
Crear publicacion de usuario inválido	Lo Permite	Devuelve null	400
Consultar el timeline de usuario no registrado	Array vacío	Array vacío	400
Followee de usuarios no registrados:	NoSuchElementException	NullPointerException	400
Followee de followee no registrado:	NoSuchElementException	NullPointerException	400
Followee de usuario no registrado:	Lo Permite	NullPointerException	400



Impacto frente al cambio



Hernan Wilkinson
@HernanWilkinson

...

En el episodio de ayer de "Diseño a la Gorra" me quedé con el sabor amargo de no haber mostrado bien el comportamiento de las distintas implementaciones respecto a modificaciones.

¿Les parece abordar nuevamente el tema pero mejor y en más detalle en el próximo episodio?

[Translate Tweet](#)

Si! veamoslo nuevamente

93.5%

No, estuvo bien así

6.5%

46 votes · Final results



Extensibilidad e Impacto Frente al Cambio

Parte B



Cambios

- Que el usuario tenga un url
 - No vamos a validar que la url sea válida
- Poder dar “like” a una publicación
 - Puede ser cualquier usuario, no solo followees



Agregar url a User



POST

/users Register a new user.

Parameters

Try it out

Name

Description

body * required

object

(body)

New user information

Example Value | Model

```
{
  "username": "string",
  "password": "string",
  "about": "string",
  "url": "string"
}
```

Parameter content type

application/json





GET

/users Return all users.

Parameters

Try it out

No parameters

Responses

Response content type

application/json



Code

Description

200

Successful operation

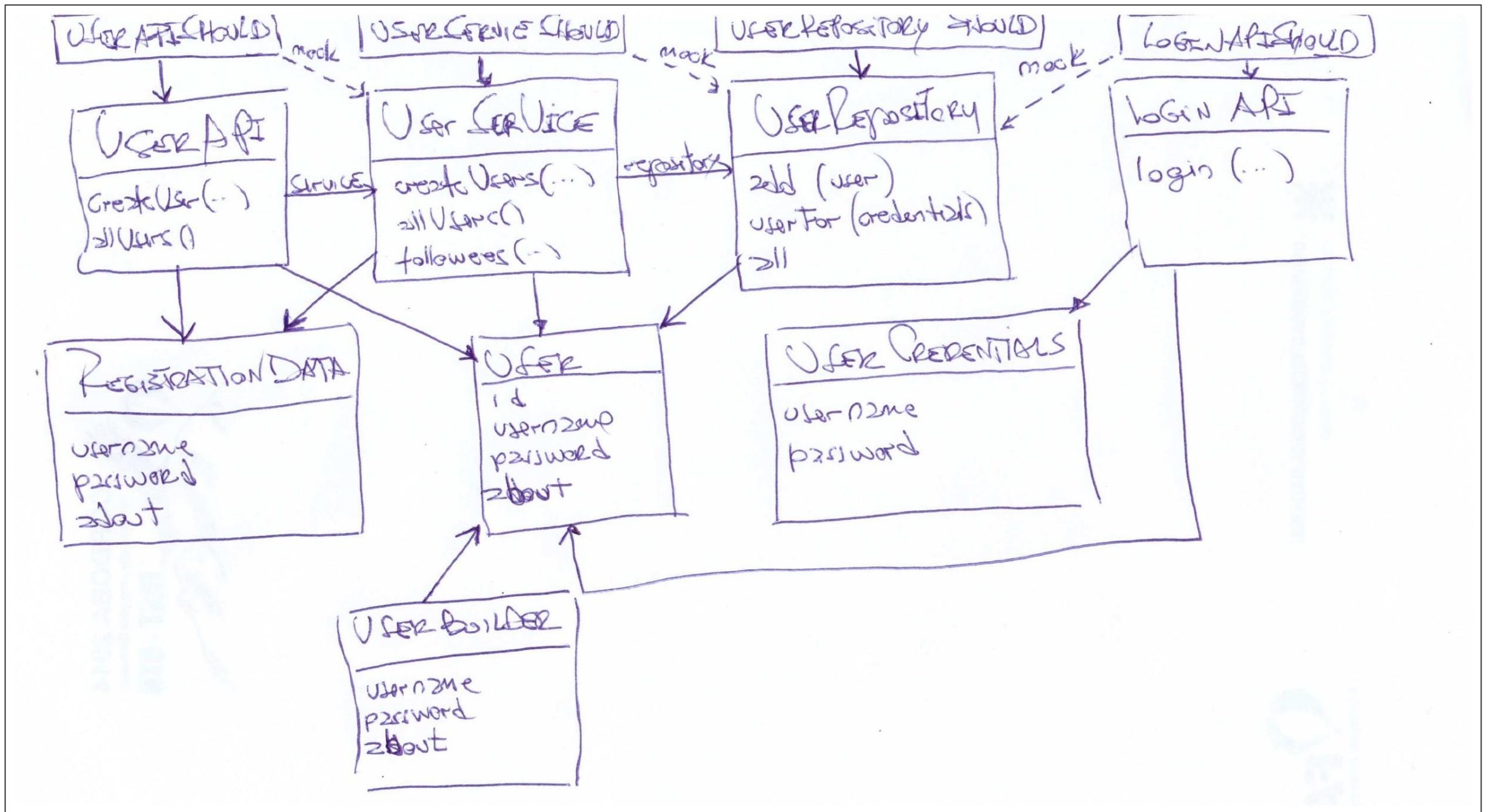
Example Value | Model

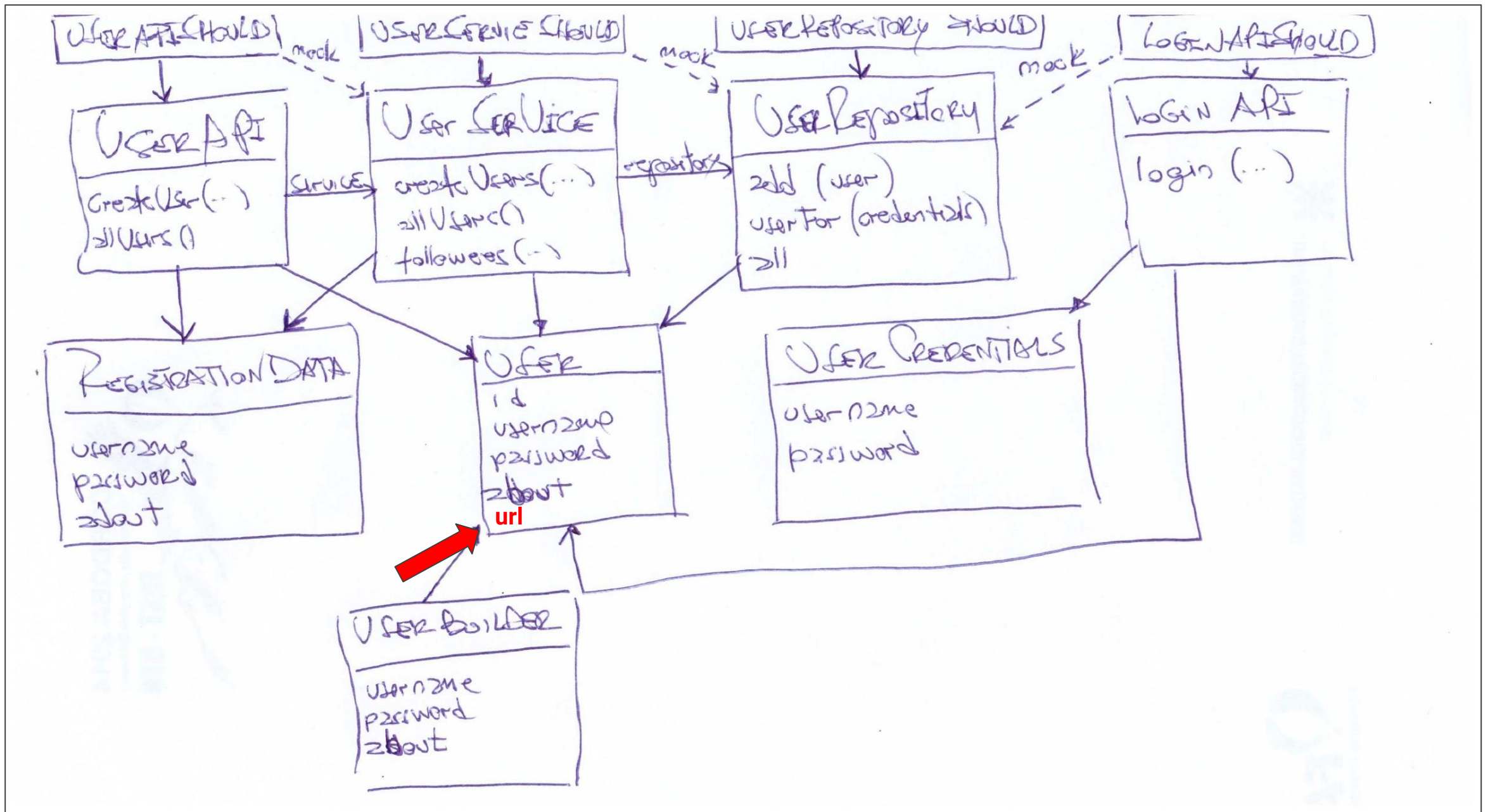
```
[
  {
    "id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "username": "string",
    "about": "string",
    "url": "string"
  }
]
```

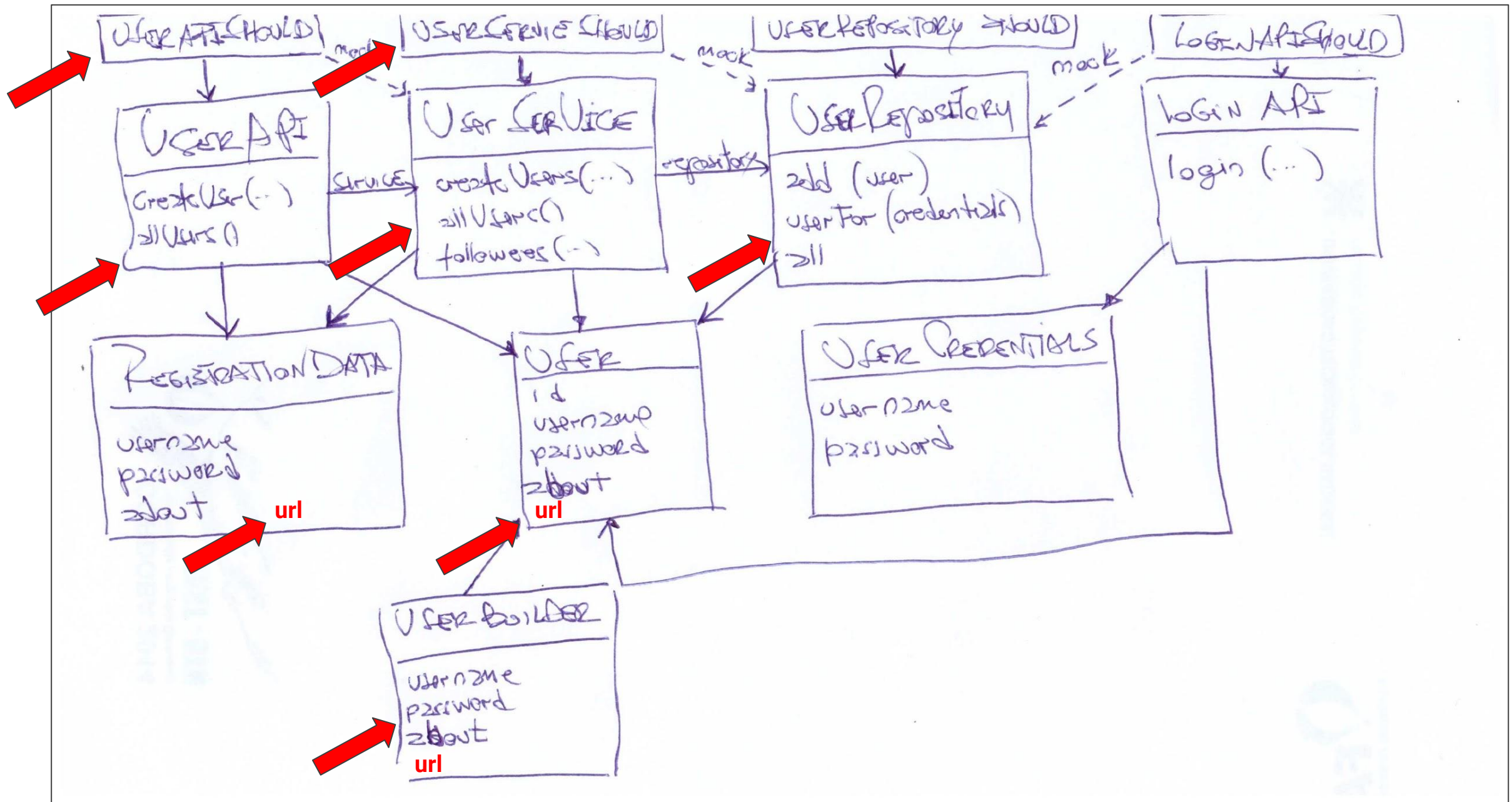


London



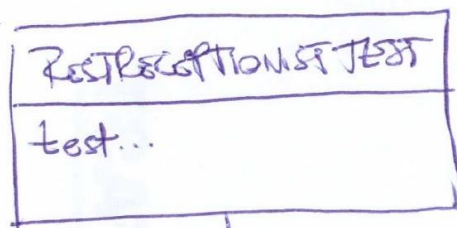




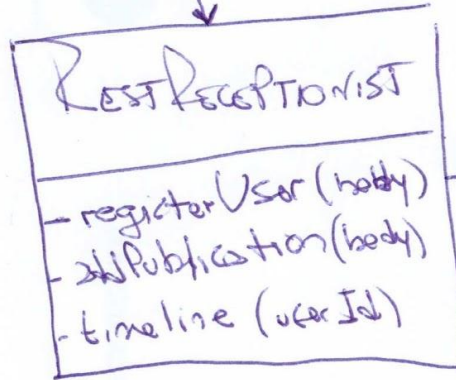


Buenos Aires

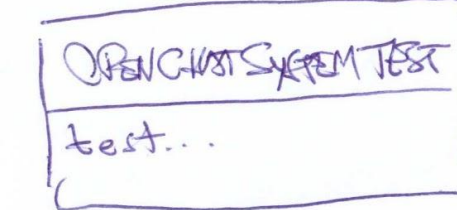




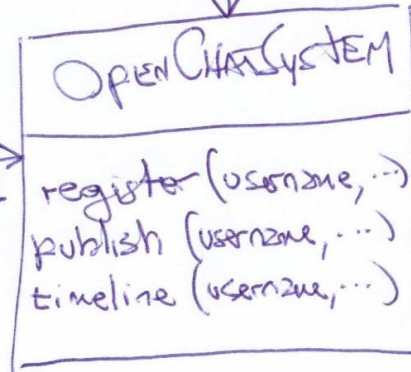
receptionist



system



system

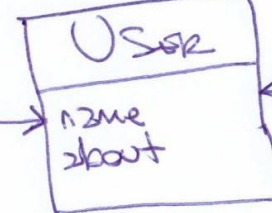
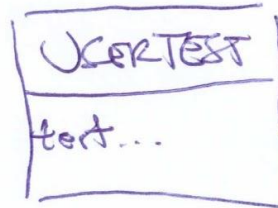


userCards



user

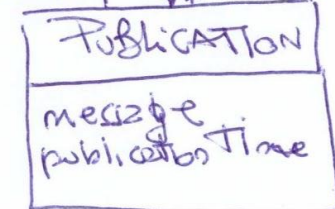
publisher

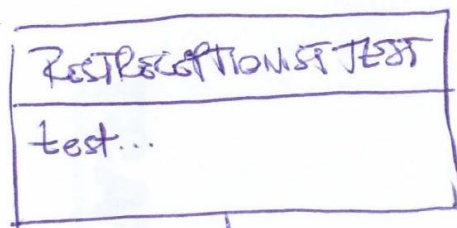


followers

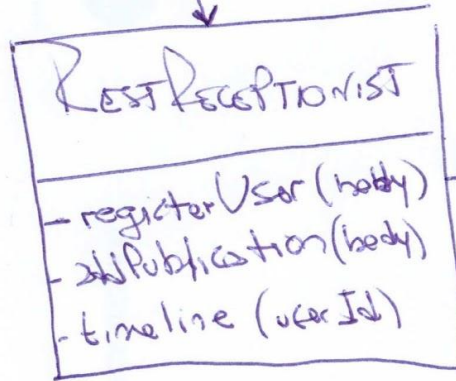
publisher

publications

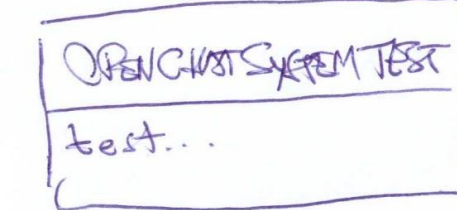




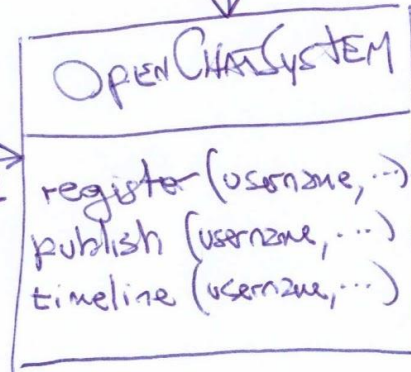
receptionist



system



system

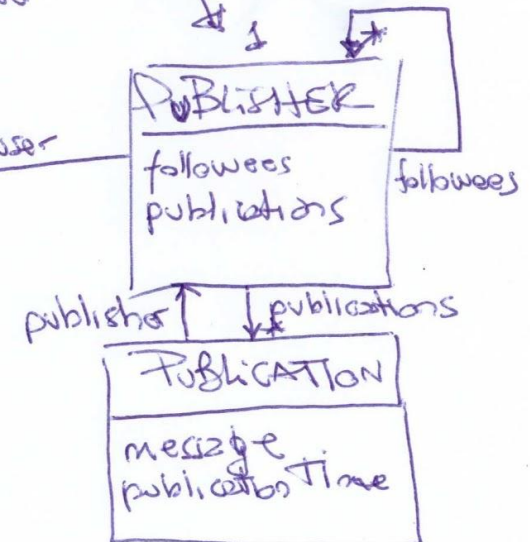
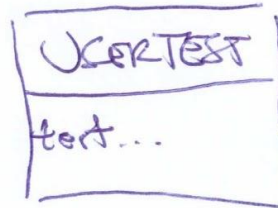


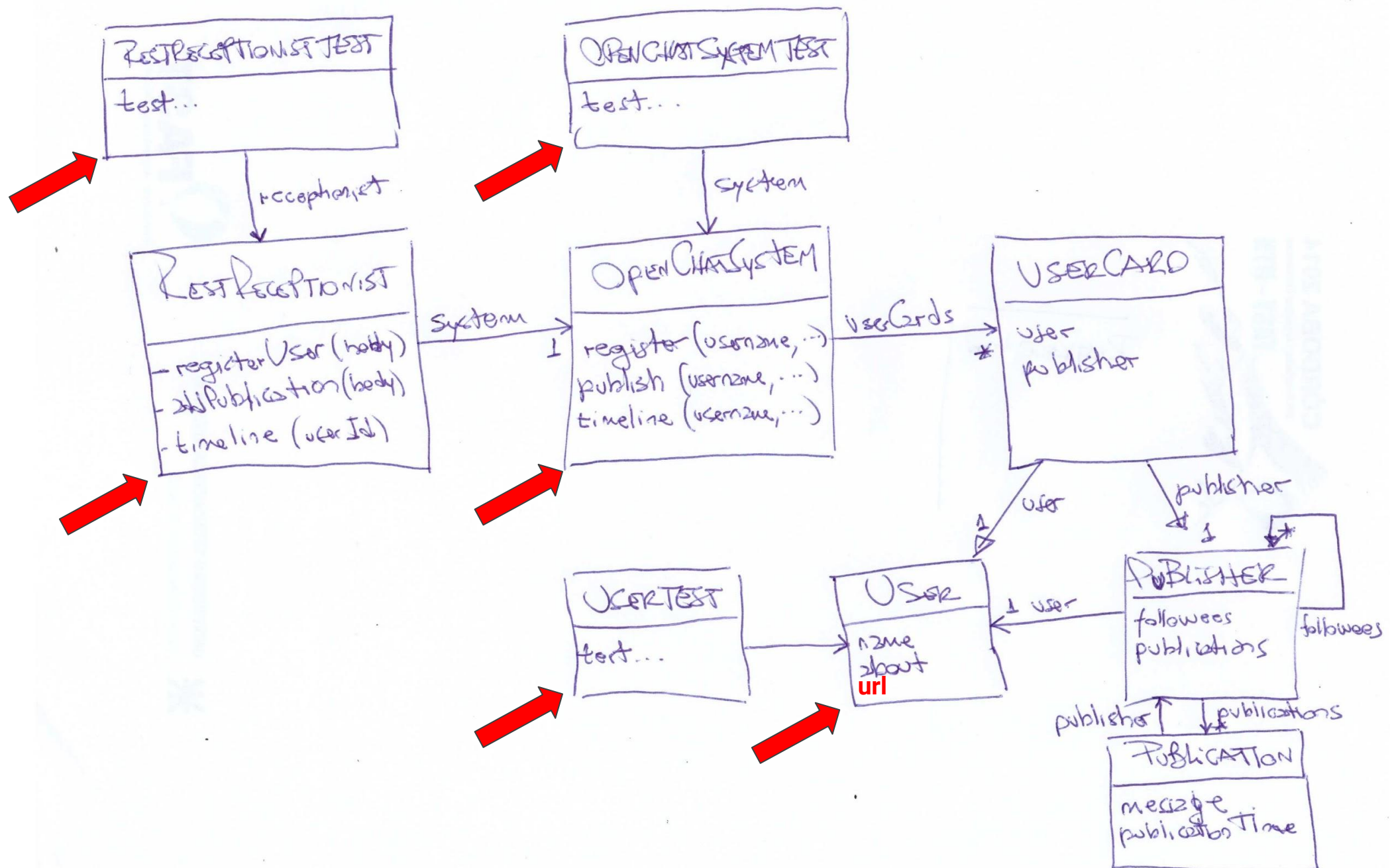
userCards

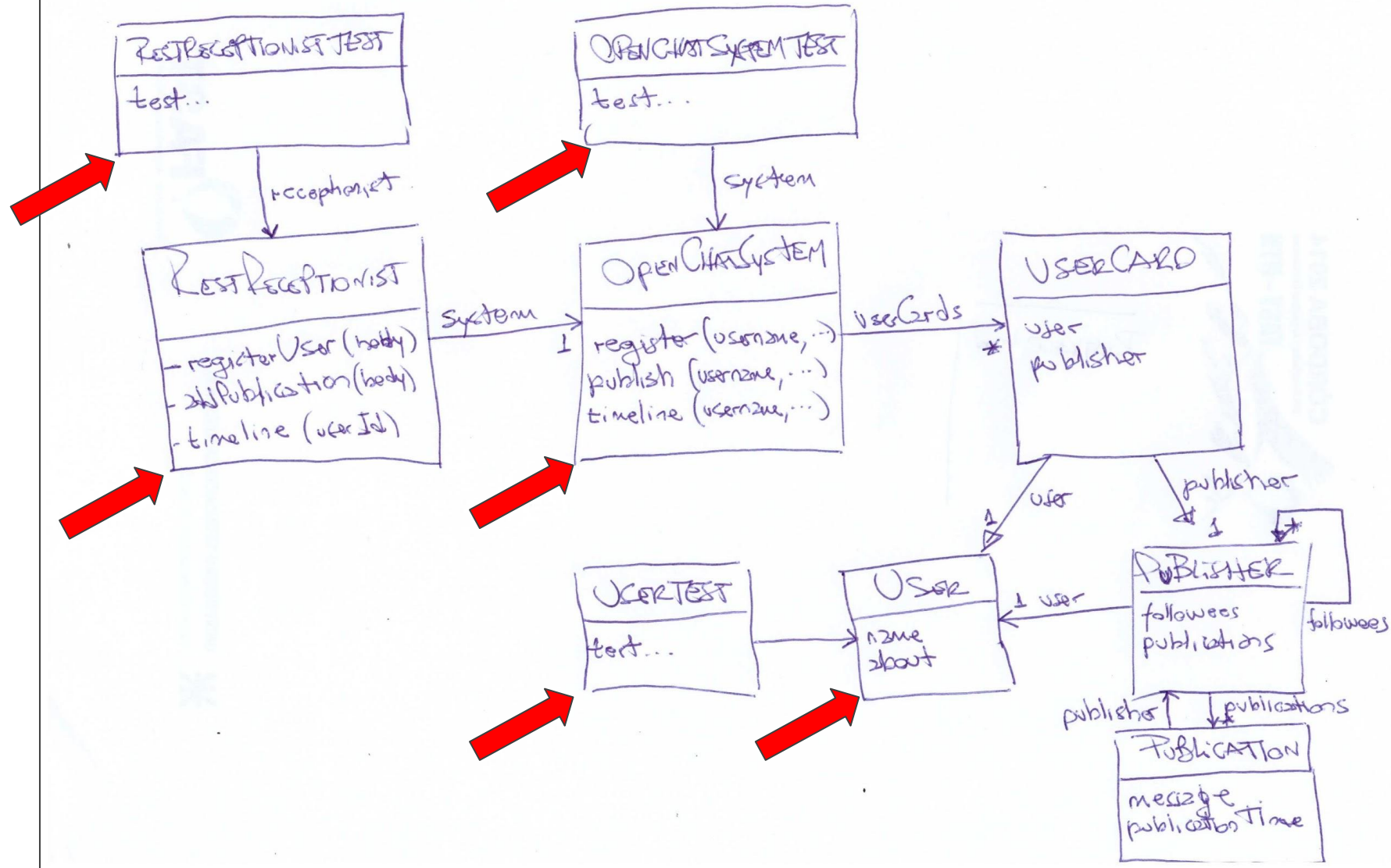


user

publisher







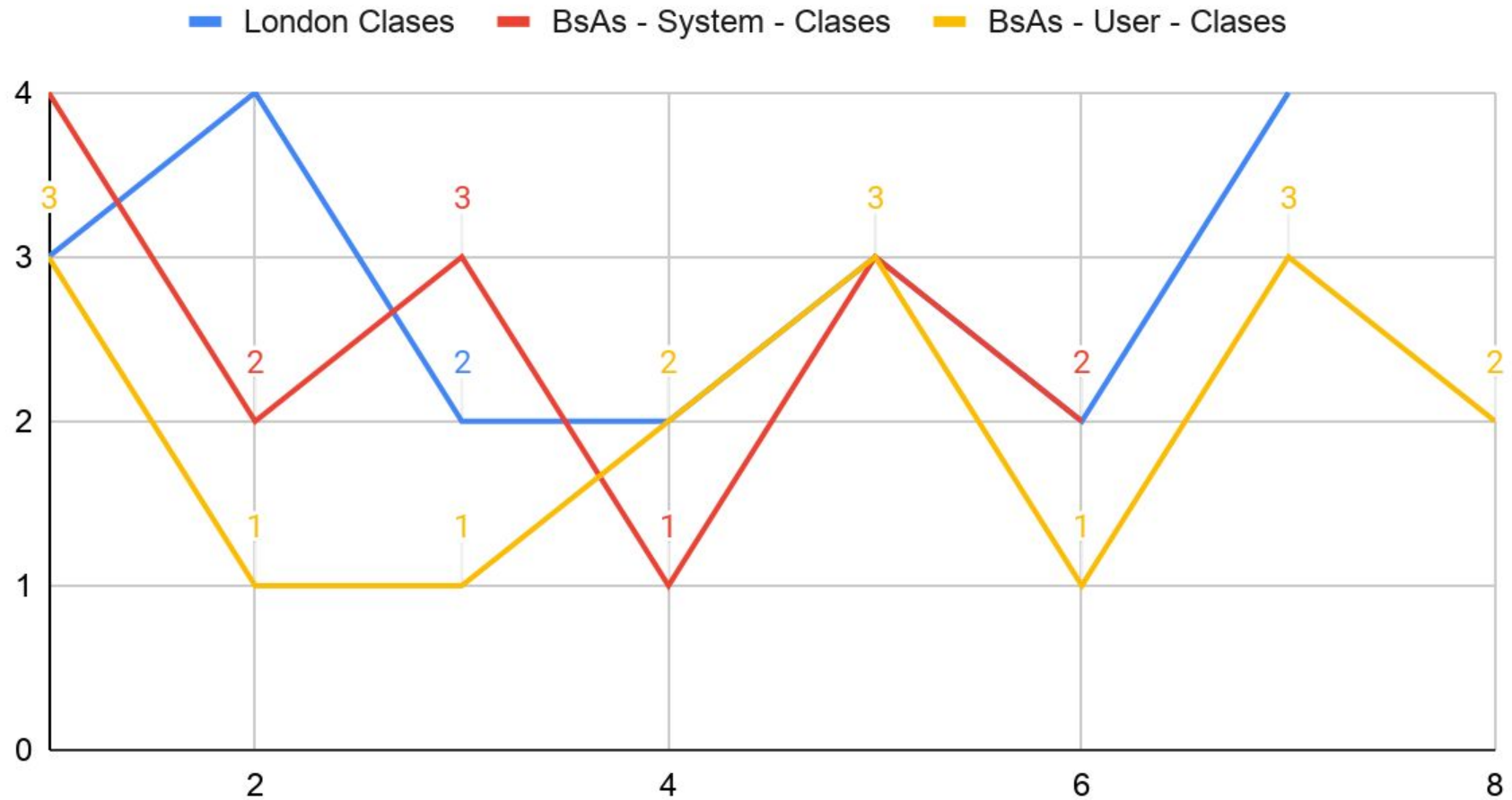
Estadísticas



		London Clases	BsAs - System - Clases	BsAs - User - Clases	
Test de UserAPIShould fallando	1	3	4	3	Test fallando en UserTest
Test de UserAPIShould pasando	2	4	2	1	Test pasando en UserTest
Resto de Test de API pasando	3	2	3	1	Test fallando en OpenChatSystem
Test de UserServiceShould que pasa de entrada	4	2	1	2	Test pasando en OpenChatSystem
Remove User.constructor	5	3	3	3	Test fallando en RestReceptionist
Remove RegistrationData.constructor	6	2	2	1	Test pasando en RestReceptionist
-Sacar hardcoded homePage	7	4		3	Remove method User.named
	8			2	Remove method OpenChatSystem.register



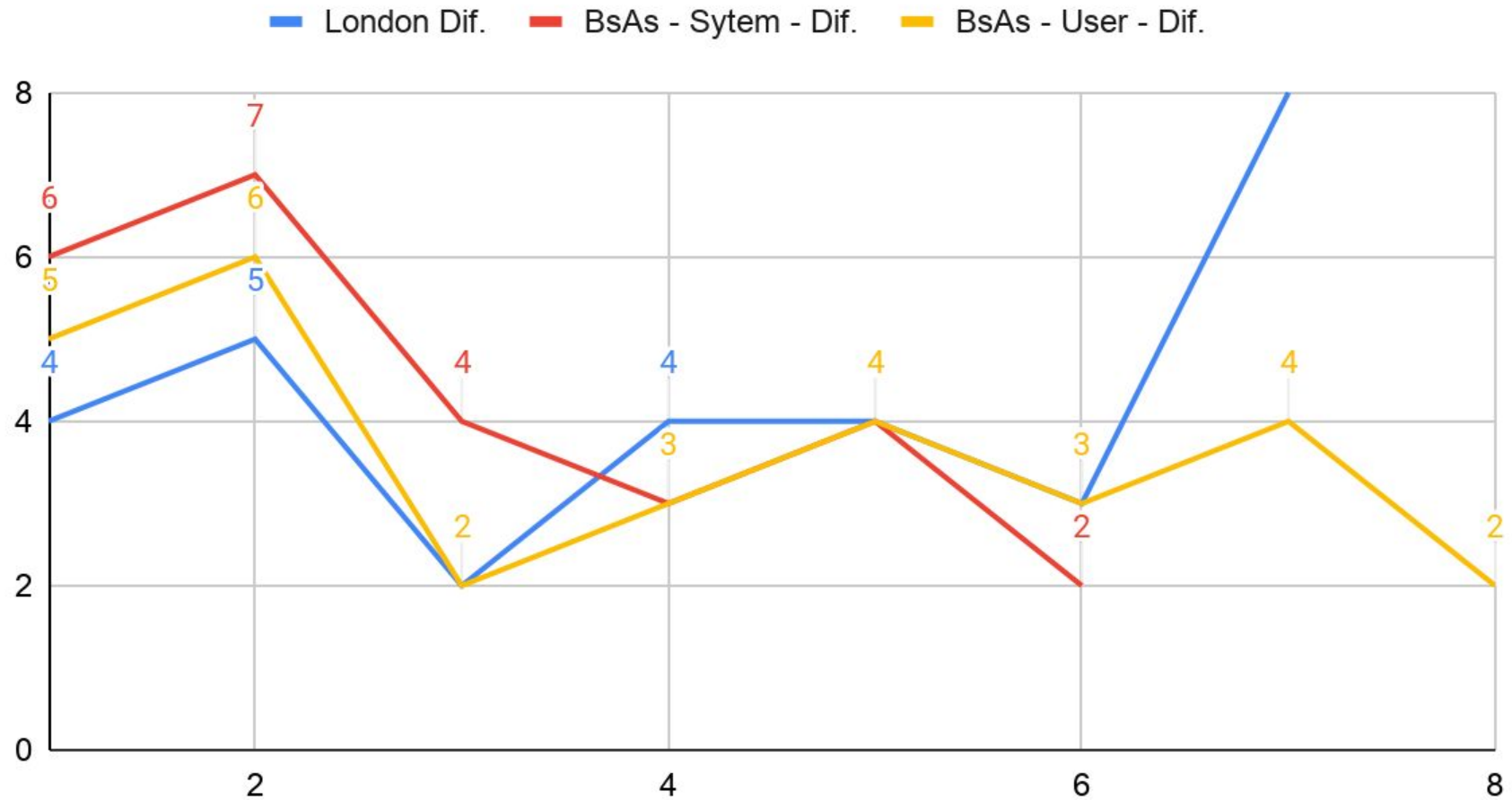
Agregar homePage - Clases



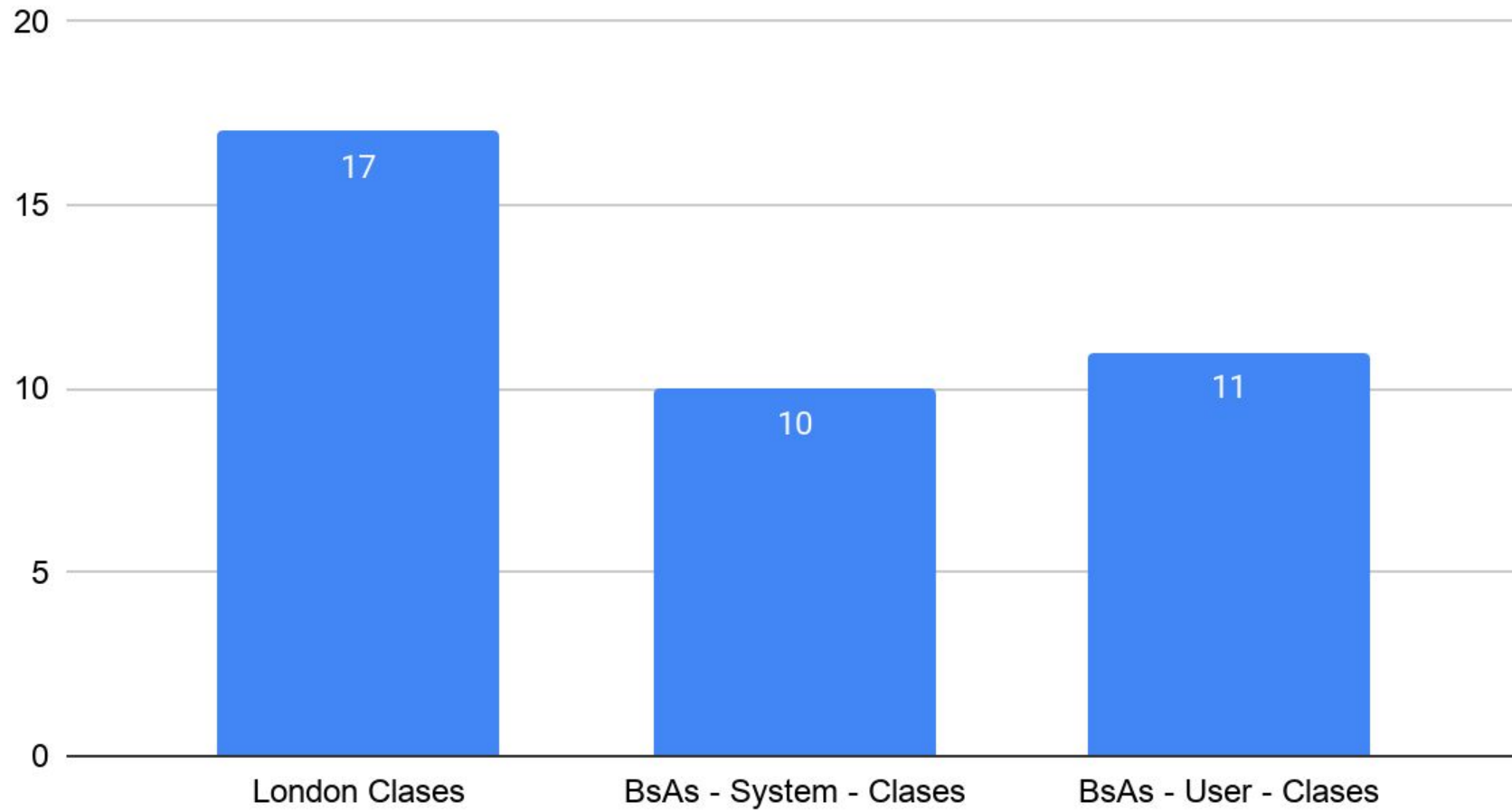
		London Dif.	BsAs - Sytem - Dif.	BsAs - User - Dif.	
Test de UserAPIShould fallando	1	4	6	5	Test fallando en UserTest
Test de UserAPIShould pasando	2	5	7	6	Test pasando en UserTest
Resto de Test de API pasando	3	2	4	2	Test fallando en OpenChatSystem
Test de UserServiceShould que pasa de entrada	4	4	3	3	Test pasando en OpenChatSystem
Remove User.constructor	5	4	4	4	Test fallando en RestReceptionist
Remove RegistrationData.constructor	6	3	2	3	Test pasando en RestReceptionist
-Sacar hardcoded homePage	7	8		4	Remove method User.named
	8			2	Remove method OpenChatSystem.register



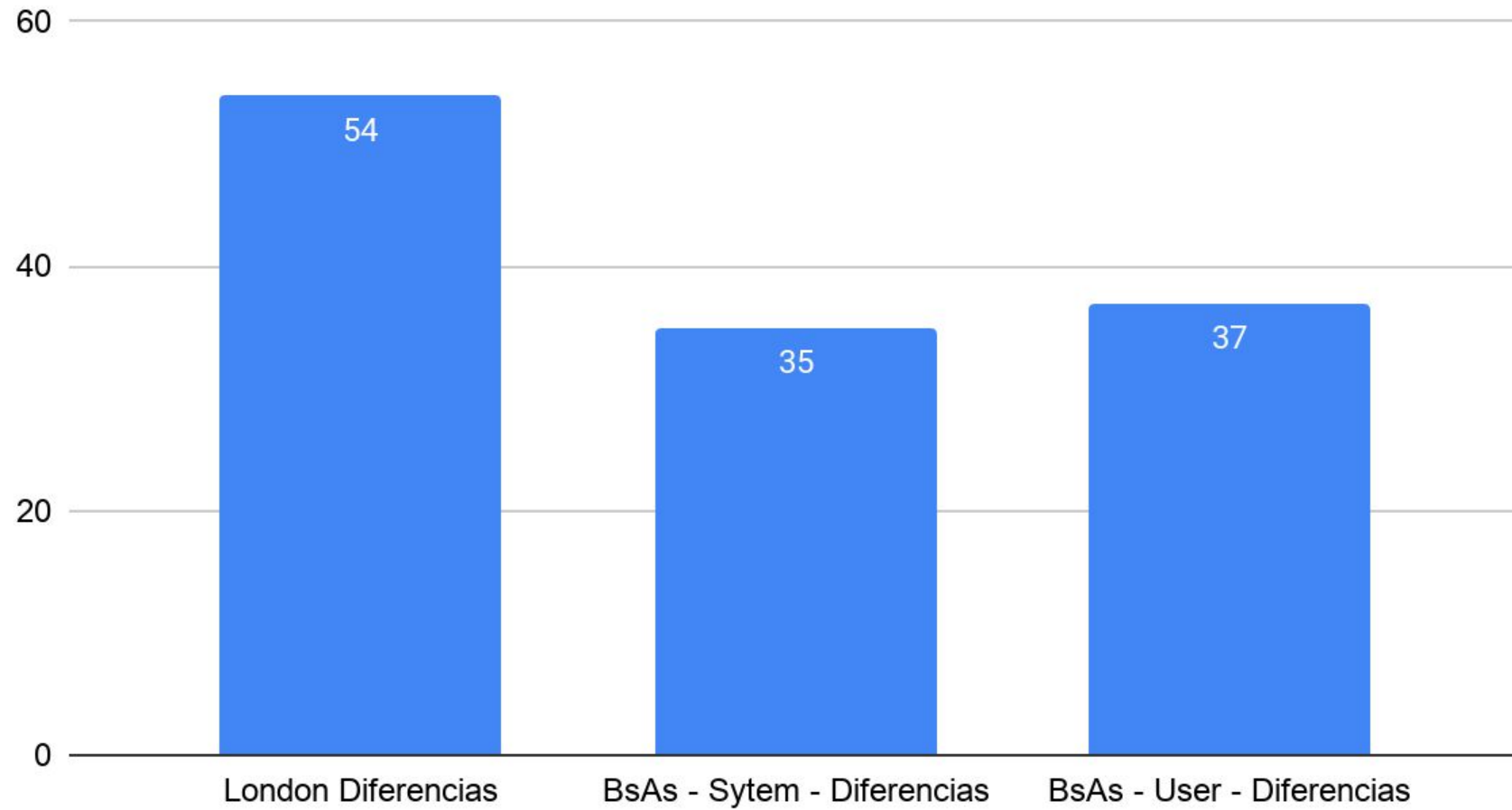
Agregar homePage - Diferencias



Rename url x homePage - Clases



Rename url x homePage - Diferencias



Like a Publication



POST

/publications/{publicationId}/like Likes a post.

Parameters

Try it out

Name

Description

publicationId * required

string(\$UUID)
(path)

ID of the publication to like.

publicationId - ID of the publication to like.

userId * required

(body)

userId that provides the like

Example Value | Model

Parameter content type

application/json



GET**/users/{userId}/timeline** Return user's timeline.**Code****Description**

200

Successful operation

Example Value | Model

```
[
  {
    "postId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "userId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "text": "string",
    "dateTime": "2018-01-10T11:30:00Z",
    "likes": 0
  }
]
```



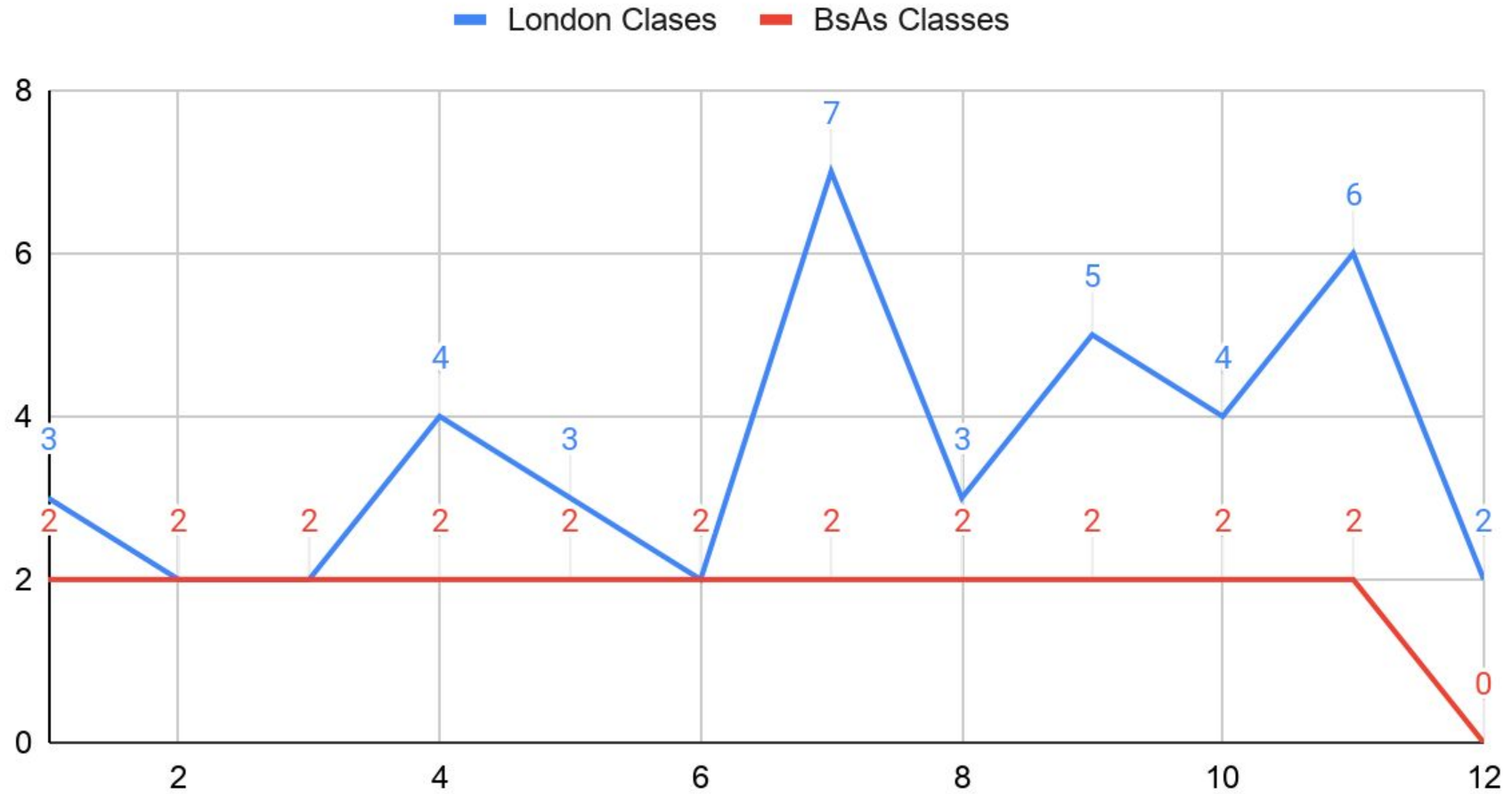
Estadísticas



		London Clases	BsAs Classes	
can_like_a_post	1	3	2	publicationsHaveNoLikesWhenCreated
like_a_post_return_number_of_likes	2	2	2	registeredUserCanLikePublication
like_post	3	2	2	canNotLikeNotPublishPublication
different_users_likes_do_count	4	4	2	likesByUserCountOnlyOnce
same_user_likes_do_not_count	5	3	2	notRegisteredUserCanNotLikePublication
return_post_identified_by_id	6	2	2	registeredUserCanPublishAppropriateMessage
throw_exception_with_invalid_post_id	7	7	2	userCanLikePublication
Likes en Post json	8	3	2	notRegisteredUserCanNotLikePublication
reject_likes_from_not_registered_user	9	5	2	canNotLikeNotPublishedPublication
throw_invalid_user_when_liking_with_invalid_user_id	10	4	2	timelineIncludesLikes
reject_likes_from_invalid_post	11	6	2	Test de aceptacion pasando
Test de aceptación pasando	12	2	0	



Agregar Likes - Clases Modificadas

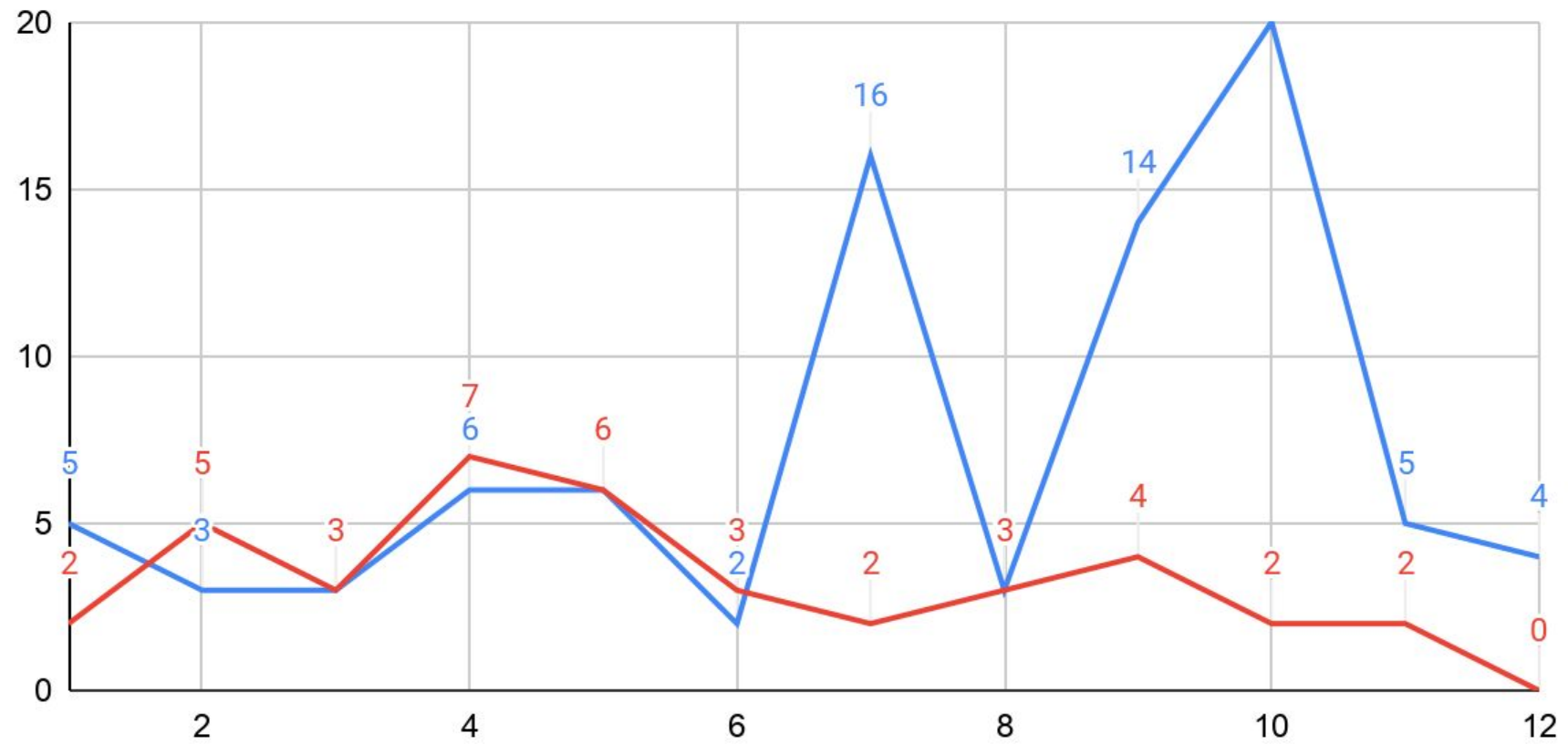


		London Diferencias	BsAs Diferencias	
can_like_a_post	1	5	2	publicationsHaveNoLikesWhenCreated
like_a_post_return_number_of_likes	2	3	5	registeredUserCanLikePublication
like_post	3	3	3	canNotLikeNotPublishPublication
different_users_likes_do_count	4	6	7	likesByUserCountOnlyOnce
same_user_likes_do_not_count	5	6	6	notRegisteredUserCanNotLikePublication
return_post_identified_by_id	6	2	3	registeredUserCanPublishAppropriateMessage
throw_exception_with_invalid_post_id	7	16	2	userCanLikePublication
Likes en Post json	8	3	3	notRegisteredUserCanNotLikePublication
reject_likes_from_not_registered_user	9	14	4	canNotLikeNotPublishedPublication
throw_invalid_user_when_liking_with_invalid_user_id	10	20	2	timelineIncludesLikes
reject_likes_from_invalid_post	11	5	2	Test de aceptacion pasando
Test de aceptación pasando	12	4	0	



Agregar Likes - Diferencias

London Diferencias BsAs Diferencias



Solución Buenos Aires

- **repo:** <https://github.com/hernanwilkinson/openchat-buenosaires>
- **agregar url:**
 - branch: empezando-desde-user
 - commit: Tests de Aceptación de url en User
 - rev. number: 129a87b7d80a33da93f6e3a6c124514793515faa
- **likes:**
 - branch: master
 - commit: >> Likes: Especificación y tests de Aceptación
 - rev. number: db911fae0dfd4867a74ade31cec903b5149f2be4



Solución London

- **repo:** <https://github.com/hernanwilkinson/openchat-london>
- **agregar url:**
 - branch: manteniendo-interface
 - commit: Test de UserAPIShould fallando
 - rev. number: 041ffa718c0c05718b05d90f1447e1940d1b61c1
- **likes:**
 - branch: default-url
 - commit: >> Likes: Especificación y tests de Aceptación
 - rev. number: da98f3c44ba08b6e8265e0f909bcbb4e255422d0



¡Lo que viene!
¡Lo que viene!



Lo que viene

- ✓ Parte 1: Modelo Inicial
- ✓ Parte 2: Model del Sistema
- ✓ Parte 3: Implementar la interfaz Rest
- ✓ Parte 4: Analizar extensibilidad y impacto frente al cambio de cada solución
- Parte 5: Hacerlo persistente + Concurrente
- Parte 6: ... suspenso :-)



Preguntas





Diseño ¡a la gorra!

¡Bienvenidos!

Durante esta serie de Webinars exploraremos *qué* significa **Diseñar Software con Objetos** y *cómo* lo podemos hacer cada vez mejor.

Te esperamos todos los Martes a las 19 Hrs GMT-3 a partir del Martes 11 de Agosto de 2020. Para poder participar tenes que **REGISTRARTE ACA.**

Trataremos muchos temas que irán desde cuestiones filosóficas como qué significa Diseñar en nuestra profesión y dónde está expresado ese Diseño, pasando por consejos y heurísticas para diseñar "mejor" para terminar con ejemplos concretos de cómo aplicar esas heurísticas en la *vida real*.

Los webinars son "*language agnostic*", o sea que no dependen de un lenguaje de programación en particular, aunque los ejemplos que usaremos estarán hechos principalmente en **Java**, **JavaScript**, **Ruby**, **Python** y mi querido **Smalltalk** cuando amerite 😊.

Todo el código y presentaciones estarán disponibles para que lo puedan usar y consultar en cualquier momento [acá](#).

¡Trae ganas de aprender y pasarla bien!

Donaciones



\$250 - Una buena 🍺

Pagar

\$500 - Menos que 🍔 + 🍕

Pagar

\$1.000 - Dos buenos 🍷

Pagar



Donate



¿Querés donar un monto variado o en otra plataforma?

<https://alagorra.10pines.com>

Muchas gracias





10 Pines

Creative Software Development



10pines.com



info@10pines.com



+54 (011) 6091-3125 / 4893-2057



Av. Leandro N. Alem 896 6° - Bs. As. - Argentina



@10pines