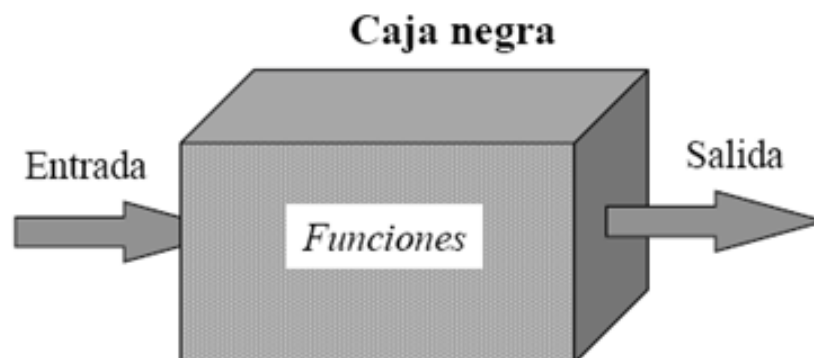
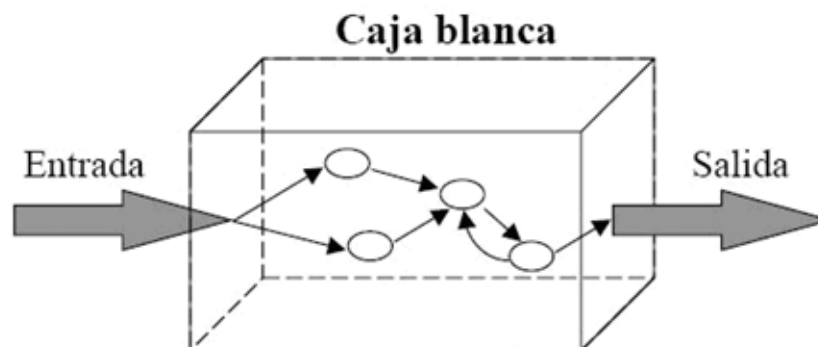


Pruebas de caja blanca y caja negra



Proyecto: Communicate

Índice:

1. Prueba de caja blanca	3
1.1. Método search()	3
1.2.Función obtain_messages (Socket.IO)	6
2. Pruebas de caja negra	7
2.1. Método search()	7

Proyecto: Communicate

1. Prueba de caja blanca

1.1. Método search()

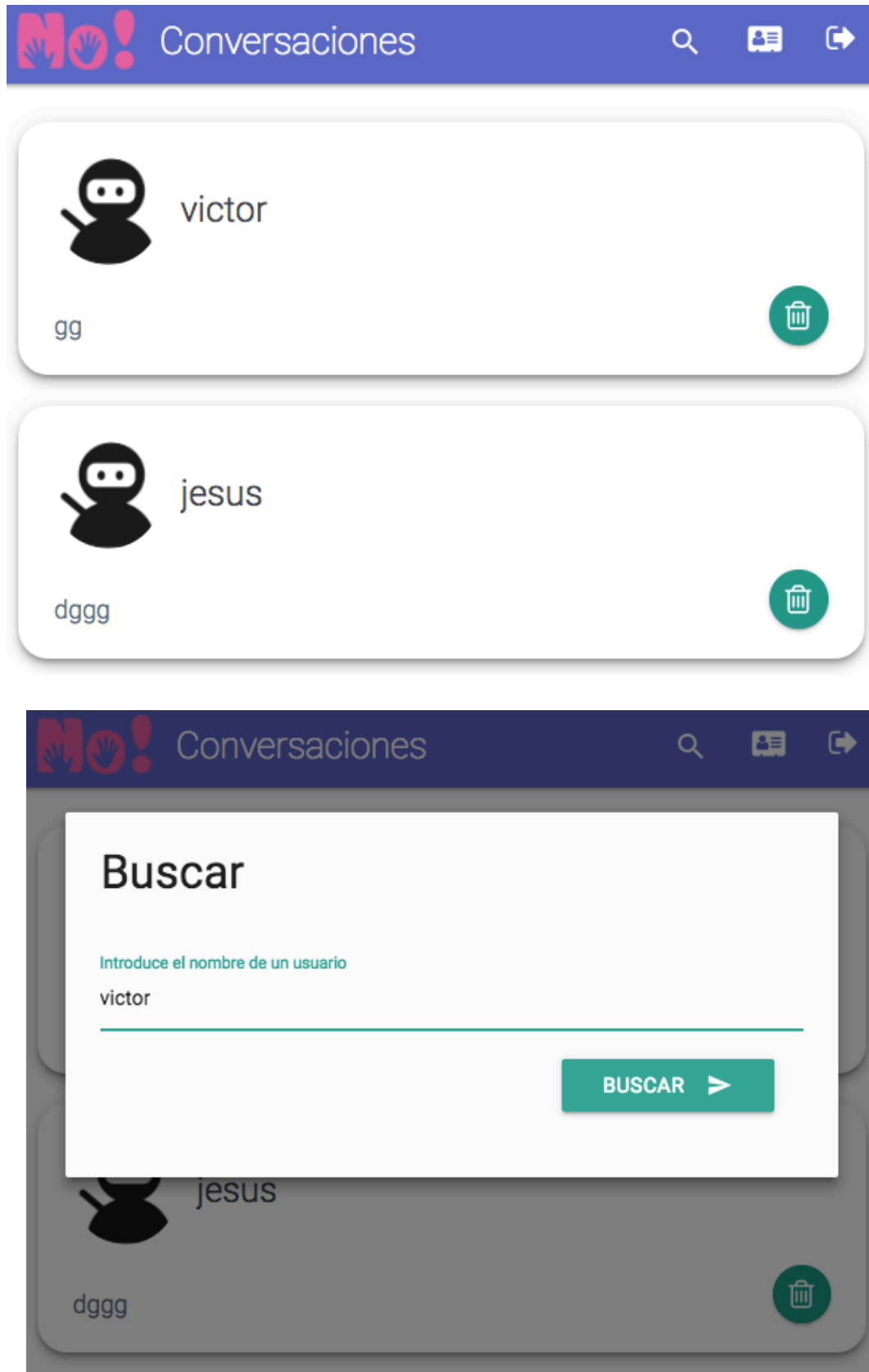
El método `search()` de `chats.js` se usa para poder filtrar los nombres de usuarios que contengan un campo de texto especificado por el usuario. Para ello primero mostramos todos los chats (para suprimir una búsqueda anterior si la hubiera), luego obtenemos un array con todas las cards de las conversaciones y después las recorremos y ocultamos las conversaciones cuyo nombre de usuario no contienen el campo de texto especificado. En caso de que no haya ninguna conversación con esos parámetros mostramos una etiqueta que indica que no existen usuarios.

Para realizar el test hemos utilizado Chrome Canary, que nos permite hacer test de cobertura, y nos muestra todos los caminos linealmente independientes del grafo de flujo de programa.

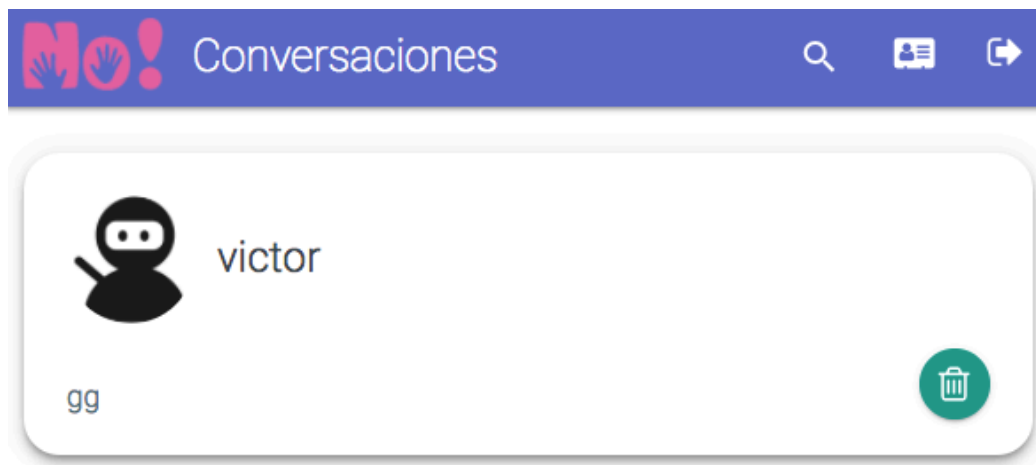
Para este fin utilizamos Chrome Canary, que muestra gráficamente (mediante colores) si efectivamente se ha visitado cada parte del código, lo que se denomina cobertura. El color **verde** indica que esa instrucción se ha ejecutado, el **amarillo** que lo ha hecho parcialmente (por ejemplo, si dicha instrucción llama a un método de otro objeto del que no se ejecutan todas sus instrucciones, en cuyo caso determina las instrucciones que no se han ejecutado – missed) y el **rojo** que no se ha ejecutado, es decir, la ejecución no pasa por ahí.

Vemos a continuación la iteración correspondiente a la función `search()`:

Proyecto: Communicate



Proyecto: Comunicate



```
33 function search(){
34
35     $('#elements>div').css("display", "block");
36
37     $('#modal_search').modal('close');
38
39     var cards = $('#elements').children();
40     var search = $('#search').val().toUpperCase();
41
42     if (search.length > 0) {
43
44         var cont = 0;
45
46         for (var i = 0; i < cards.length; i++) {
47
48             var nameChat = cards[i].getElementsByClassName("card-title text_title_
49
50             if (!nameChat.includes(search)){
51
52                 $(cards[i]).css("display", "none");
53
54             }else {
55
56                 cont++;
57
58             }
59
60         }
61
62         if (cont == 0) {
63
64             noResult = '<center><h4 class="no_result">No se encontró ningún chat</
65             $("#elements").append(noResult);
66
67         }
68
69     }
70
71 }
72
```

Proyecto: Comunicate

1.2.Función obtain_messages (Socket.IO)

Esta función es un manejador de señales para socket.io, lo que hace es permanecer escuchando hasta que recibe la señal `obtain_messages`. Cuando recibe esta señal ejecuta la función de su interior con parámetro `chats` que contiene un JSON con una lista de los chats abiertos para ese usuario. Esta función se usa tanto para cargar los chats, como para recargarlos en caso de que haya nuevos chats.

Lo primero que hace es borrar todos los chats que haya anteriormente, después recorre la lista de chats y los va pintando uno a uno.

Al igual que para el otro método presentamos primero los test que incluyen los casos de prueba para hacer que todo camino linealmente independiente del grafo de flujo se recorra al menos una vez.

```
1 var socket = io.connect();
2
3 socket.on('obtain_messages', function(chats) {
4
5     //alert(JSON.stringify(chats));
6
7     $("#elements").empty();
8
9     for (var i = 0; i < chats.data.listMessages.length; i++) {
10
11         chat = chats.data.listMessages[i];
12
13         var data = '<div id="card-'+chat.chatID+'" class="col s12 m6 l4"><div class="card circular_b
14
15         $("#elements").append(data);
16     }
17
18 }));
```

Proyecto: Communicate

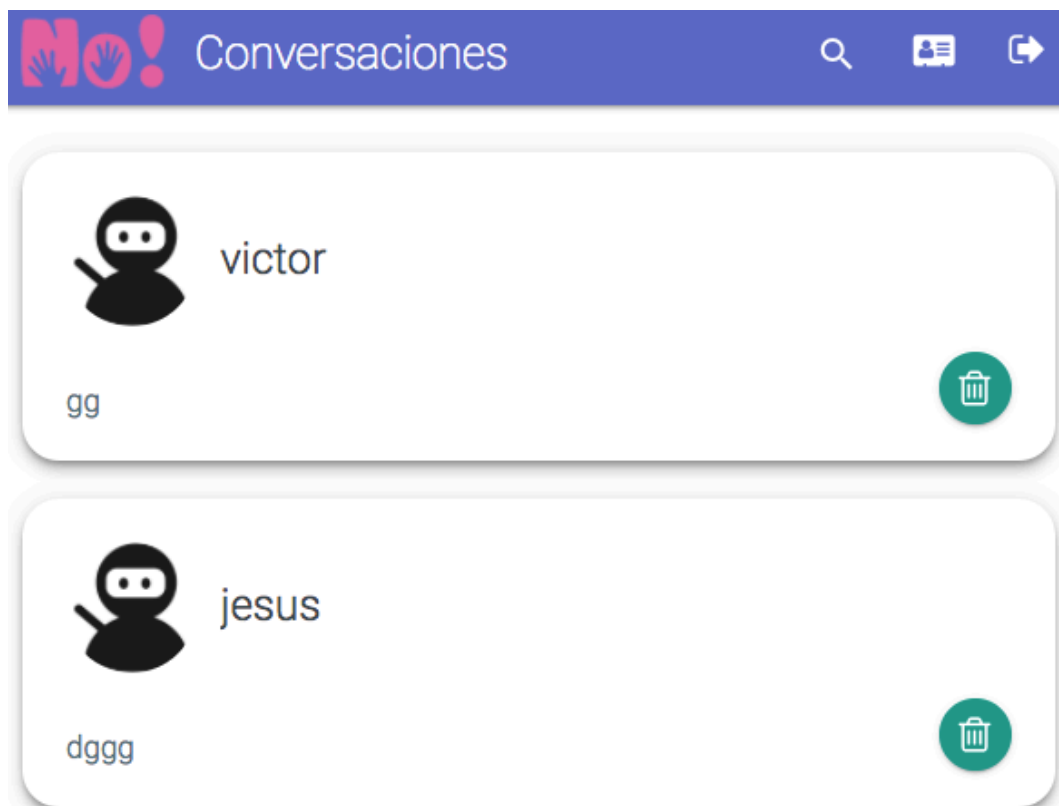
2. Pruebas de caja negra

2.1. Método search()

Procederemos a analizar el mismo método por lo tanto no volveremos a explicar su funcionalidad.

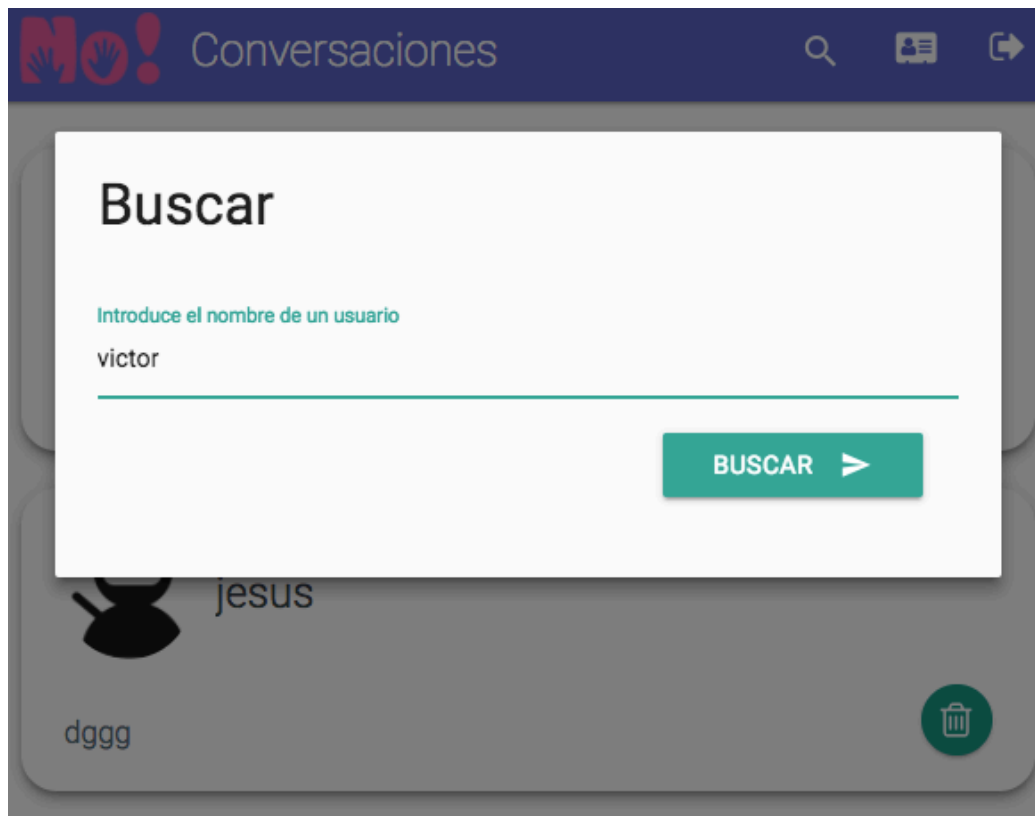
Clases validas	Clases no validas
Cualquier nombre de usuario presente en los chats	Cualquier nombre de usuario no presente en los chats

Primera ejecución para las clases validas:

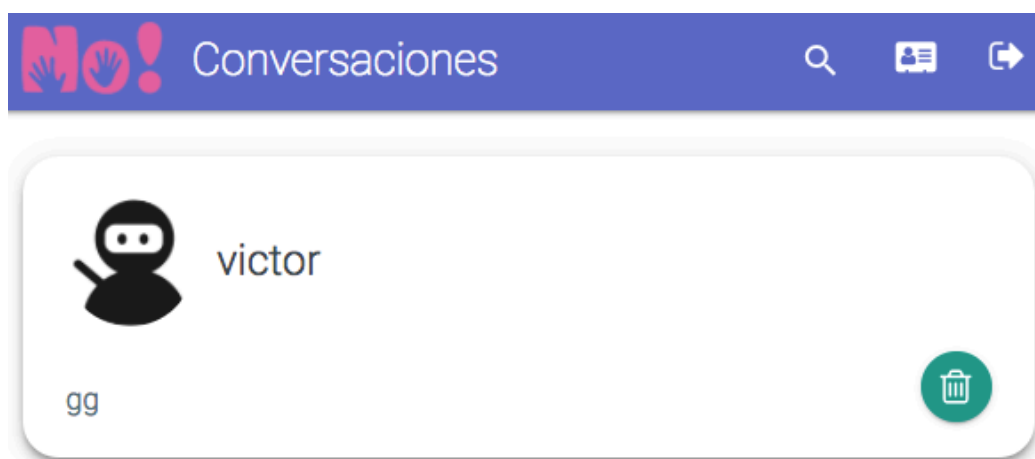


Proyecto: Communicate

Coge el texto incluido dentro del input



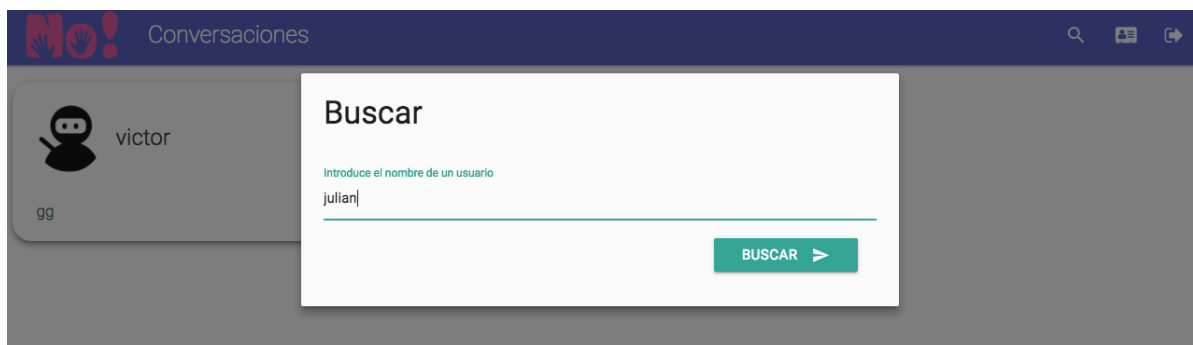
Coge todas las card donde están contenidos los chats y las recorrer eliminando las que no contienen el parámetro introducido.



Proyecto: Communicate

Primera ejecución para las clases no validas

Buscamos un nombre no contenido en ningún chat



Se muestra el mensaje “No se encontró ningún chat”



No se encontró ningún chat

Las pruebas de caja negra muestran todas las entradas factibles y no.