

TALLER DE SISTEMAS DE COMPUTACIÓN 2016-2

NWJS + API REST

INTEGRANTES:
ROBERTO FUENTES
CARLOS JAUREGUI
PATRICIO SARD

PROFESOR:
EUGENIO CANALES

CREACIÓN DE LA APLICACIÓN





COMENZANDO

- Comenzamos la aplicación mostrando al usuario 2 opciones para comenzar: La de registro y la de login.





REGISTRO – VISTA

- ▶ Si el usuario no tiene una cuenta, lo primero que se deberá hacer es registrarse. Para esto, se le pide su nombre completo, su nombre de usuario en la aplicación, su correo y finalmente una contraseña. Se deben llenar todos los campos, además de colocar un mail con la estructura *mail@dominio.cl*.

The screenshot shows a web browser window with the title "Laboratorio 3 TSC 2016". The main content area is dark gray and contains a white rounded rectangle with the title "Registro de usuario". Below the title are four white input fields stacked vertically, labeled "User Name", "Full Name", "Email", and "Password". At the bottom of the form are two blue buttons: "Registrarse" and "Volver".



REGISTRO - CÓDIGO

- ▶ Para realizar esto, por dentro se guarda la información de cada campo que el usuario llenó. Luego, se verifica si todos los campos están llenos, y si el mail es válido, para luego mandar una consulta a la API con los datos ingresados. Finalmente, si la respuesta por parte de la API es exitosa, se mandara al menú principal de la aplicación.

```
$('#button-register').click(function() {  
    var data = {}  
    data.full_name = $('#full_name').val();  
    data.user_name = $('#user_name').val();  
    data.email = $('#email').val();  
    data.password = $('#password').val();  
  
    if(validarEmail($('#email').val())){  
        if($('#full_name').val() != '' || $('#user_name').val()  
        | ($('#email').val() != '' || ($('#password').val() != '')){  
            users.post(data);  
        }  
        else{  
            alert("Debe llenar todos los campos.");  
        }  
    }  
});
```

```
post: function(data){  
    $.ajax(  
        {  
            url : "http://localhost:9999/users",  
            data : JSON.stringify({  
                full_name: data.full_name,  
                user_name: data.user_name,  
                email: data.email,  
                password: data.password  
            }  
        ),  
            contentType : "application/json; charset=UTF-8",  
            dataType : "json",  
            type : "POST",  
            processData : false,  
            timeout : 30000,  
            success : function(json) {  
                //console.log(json);  
                alert("Usuario Creado");  
                location.reload();  
                window.location.href = 'notes.html';  
            },  
            error : function(xhr) {  
                //console.log(xhr);  
                alert("Error al crear el usuario");  
            },  
            complete : function(xhr, status) {  
            }  
        }  
    );  
}
```



LOGIN - VISTA

- ▶ Aquí mostramos 2 opciones para el usuario. Una es el login directo, usando su *mail* y *password*. En caso de no tenerla, también esta la opción de poder registrarse.

The screenshot shows a web browser window with the title "Laboratorio 3 TSC 2016". The main content area is dark gray and contains a centered login form. The form has a title "Login" in white text. Below the title are two white input fields: "Email" and "Password". At the bottom of the form are two blue buttons: "Login" and "Registrarse".



LOGIN - CÓDIGO

- ▶ Aquí verificamos si todos los campos están llenos, guardando las variables de *email* y *password* que posteriormente serán usadas para las consultas en la API. Posteriormente, se manda una consulta de tipo **POST** a la API con los datos *encodeados* en *base64*. Si la consulta es exitosa, se mandara al menú principal.

```
var auth = {
  post: function(data){
    $.ajax(
      {
        url : "http://localhost:9999/users/login",
        data : JSON.stringify({
          credentials: btoa(data.email + ':' + data.password)
        }),
        contentType : "application/json; charset=UTF-8",
        dataType : "json",
        type : "POST",
        processData : false,
        timeout : 30000,
        success : function(json,data) {
          location.reload();
          window.location.href = 'notes.html';
        },
        error : function(xhr) {
          console.log(xhr);
          alert("Error al logearse");
        },
        complete : function(xhr, status)
        {}
      }
    );
  }
};
```

```
$('#button-login').click(function() {
  data = {};
  data.email = $('#email').val();
  data.password = $('#password').val();

  if(validarEmail($('#email').val())){
    if(data.password != '' && data.email != ''){
      window.localStorage.setItem('email_user', $('#email').val());
      window.localStorage.setItem('password_user', $('#password').val());
      auth.post(data);
    }
  }
  else{
    alert("Debe rellenar todos los campos.");
  }
});
```



NOTAS DE USUARIO - VISTA

- Una vez logueado en la aplicación, podremos acceder al menú principal de notas, donde se nos mostraran todas las notas que hemos creado. Además, podremos crear una nota, ver los detalles de una nota en particular, y borrar alguna nota que el usuario escoja.

Aplicación de Notas Grupo 2

TSC 2016-2 Bienvenido: carlos.jsuregui@alumnos.usm.cl Recargar vista Logout

Notas

Etiquetas

Crear Nota

Notas

Show 10 entries Search:

Titulo	Descripción	Fecha de creación	Detalles	Eliminar
Cartaman	Cartaman de base de datos el miercoles a las 8:40.	2016-11-16T02:36:13.270Z	Ver	
Laboratorio	Laboratorio de fisica a las 11:30 de la mañana.	2016-11-16T02:36:22.250Z	Ver	

Showing 1 to 2 of 2 entries Previous 1 Next

Crear Nota

Titulo

Descripcion

Crear



NOTAS DE USUARIO - CÓDIGO

- Lo que hacemos es pedirle a la API que nos mande todas las notas del usuario. Si la petición es exitosa, cargamos en una tabla las etiquetas mostrando el título, descripción y cuando se creó cada nota. Al hacer click al botón `#new-note-button`, mandamos una solicitud **POST** con los datos de la nota.

```
var notesView = {
  init: function() {
    navbarView.init();
    notes.get_all();

    $('#new-note-button').click(function() {
      var data = {};
      data.title = $('#title').val();
      data.description = $('#description').val();
      notes.post(data);
    })
  },

  buildTable: function(data) {
    var table = $('#tabla-notes').DataTable({
      data: data,
      columns: [
        { title: "Título ", data: "title"},
        { title: "Descripción", data: "description"},
        { title: "Fecha de creación", data: "created_at"},
        { title: "Detalles", data: null, "defaultContent": "<div class='text-center'>"},
        { title: "Eliminar", data: null, "defaultContent": "<div class='text-center'>"},
      ]
    });
  }
};
```

1

```
var notes = {
  get_all: function() {
    $.ajax({
      url: "http://localhost:9999/notes/",
      data: {},
      type: "GET",
      dataType: "json",
      beforeSend: function(xhr) {
        token = btoa(window.localStorage.getItem('email_user') + ':' + window.localStorage.getItem('password_user'));

        xhr.setRequestHeader("Authorization", token);
      },
      success: function(json) {
        notesView.buildTable(json);
      },
      error: function() {
        alert("Error al obtener los datos de las notas.");
      },
      complete: function() {

      }
    });
  },
};
```

3

```
$('#tabla-notes tbody').on('click', '.btn-success', function() {
  var data = table.row($(this).parents('tr')).data();
  window.location.href = "tags_from_note.html?id=" + data.id;
});
```

```
$('#tabla-notes tbody').on('click', '.btn-danger', function() {
  var data = table.row($(this).parents('tr')).data();
  window.sessionStorage.setItem("selected_note", data.id);
  $('#note-delete-modal').modal('show');
});
```

2

```
$('#confirm-note-delete').click(function() {
  var erased_note = window.sessionStorage.getItem("selected_note");
  notes.delete(erased_note);
});
```



AUTENTICACIÓN

- Es importante que después de loguearse, antes de todas las consultas que mandaremos a la API debemos enviar un token a la API, enviando de forma encodeada en base64 el *email* y *password* previamente guardados al momento de entrar en el menú de la aplicación.

```
var notes = {
  get_all: function() {
    $.ajax({
      url: "http://localhost:9999/notes/",
      data: {},
      type: "GET",
      dataType: "json",
      beforeSend: function(xhr) {
        token = btoa(window.localStorage.getItem('email_user') + ':' + window.localStorage.getItem('password_user'));
        xhr.setRequestHeader("Authorization", token);
      },
      success: function(json) {
        notesView.buildTable(json);
      },
      error: function() {
        alert("Error al obtener los datos de las notas.");
      },
      complete: function() {
      }
    });
  },
};
```



CREACIÓN DE NOTAS – CÓDIGO

- Aquí guardamos el título y descripción de la nota que el usuario ingresa, mandando esto al método post de la variable *notes*. Luego guardamos en formato *JSON* las variables que mandaremos, y enviamos una consulta de tipo POST a la API.

```
post: function(data) {
  $.ajax({
    url: "http://localhost:9999/notes/",
    data: JSON.stringify({
      title: data.title,
      description: data.description
    }),
    contentType: "application/json; charset=UTF-8",
    dataType: "json",
    type: "POST",
    processData: false,
    timeout: 30000,
    beforeSend: function(xhr) {
      token = btoa(window.localStorage.getItem('email_user') + ':' + window.localStorage.getItem('password_user'));

      xhr.setRequestHeader("Authorization", token);
    },
    success: function() {
      alert("Nota creada con éxito.");
      location.reload();
    },
    error: function() {
      alert("Error al crear la nota.");
    },
    complete: function() {
    }
  });
},
```

```
var notesView = {
  init: function() {
    navbarView.init();
    notes.get_all();

    $('#new-note-button').click(function() {
      var data = {};
      data.title = $('#title').val();
      data.description = $('#description').val();
      notes.post(data);
    });
  },

  buildTable: function(data) {
    var table = $('#tabla-notes').DataTable({
      data: data,
      columns: [
        { title: "Título ", data: "title"},
        { title: "Descripción", data: "description"},
        { title: "Fecha de creación", data: "created_at"},
        { title: "Detalles", data : null, "defaultContent": "<div class='text-center'><button class='btn"},
        { title: "Eliminar", data : null, "defaultContent": "<div class='text-center'><button class='btn"
      ]
    });
  }
};
```




BORRAR NOTA – VISTA

- ▶ Para borrar una nota, seleccionamos la nota que deseamos eliminar, haciendo click en el icono rojo con un basurero. Posteriormente, la aplicación nos preguntara si estamos seguros de eliminar la nota. Si nuestra respuesta es afirmativa, la nota se borrara, mostrando un mensaje de éxito.

Titulo	Descripción	Fecha de creación	Detalles	Eliminar
Certamen	Certamen de base de datos el miercoles a las 3:40.	2016-11-16T02:36:13.270Z	Ver	

Eliminar notas

¿Seguro que desea eliminar esta nota?

No

Sí, eliminar

Eliminar notas

¿Seguro que desea eliminar esta nota?

Nota borrada con éxito.

Aceptar



BORRAR NOTA – CÓDIGO

- Al seleccionar el botón de borrar, estaremos mandando al método delete la id de la nota que queremos borrar. Posteriormente, mandamos una petición de tipo DELETE a la API, con la id de la nota a eliminar.

```
delete: function(id) {
  $.ajax({
    url: "http://localhost:9999/notes/" + id,
    contentType: "application/json; charset=UTF-8",
    dataType: "json",
    type: "DELETE",
    processData: false,
    timeout: 30000,
    beforeSend: function(xhr) {
      token = btoa(window.localStorage.getItem('email_user') + ':' + window.localStorage.getItem('password_user'));

      xhr.setRequestHeader("Authorization", token);
    },
    success: function() {
      alert("Nota borrada con éxito.");
      location.reload();
    },
    error: function() {
      alert("Error al borrar la nota.");
    },
    complete: function() {
    }
  });
},
```

```
$('#tabla-notes tbody').on('click', '.btn-success', function() {
  var data = table.row($(this).parents('tr')).data();
  window.location.href = "tags_from_note.html?id=" + data.id;
});

$('#tabla-notes tbody').on('click', '.btn-danger', function() {
  var data = table.row($(this).parents('tr')).data();
  window.sessionStorage.setItem("selected_note", data.id);
  $('#note-delete-modal').modal('show');
});
```

```
$('#confirm-note-delete').click(function() {
  var erased_note = window.sessionStorage.getItem("selected_note");
  notes.delete(erased_note);
});
```



DETALLE DE UNA NOTA – VISTA

- En el detalle de una nota, podremos ver la información de una nota, y además podremos modificarla (los campos se auto-rellenarán con los datos de la nota). Por otra parte, podremos ver las etiquetas asignadas a esta nota.

Modificar nota

Título

Certamen

Descripción

Certamen de base de datos el mierccles a las 3:40.

Modificar

Certamen

Modificar nota

Etiquetas Asignadas

Show 10 entries Search:

Título	Descripción	Fecha de creación	Detalles	Eliminar
bd	Base de datos.	2016-11-16T03:00:56.067Z	Ver	

Showing 1 to 1 of 1 entries Previous 1 Next



DETALLE DE UNA NOTA – CÓDIGO

- ▶ Para guardar la información de una nota y pasarla de una vista a otra, creamos una función `setData`, donde enviamos el título y descripción de la nota, y los llenamos mediante la función `getElementById`. Además, con la función `buildTable`, creamos una tabla mostrando todas las etiquetas asignadas a esta nota.

```
var tagsFromNoteView = {
  init: function(id) {
    navbarView.init();
    notes.get_unique(id);
    tags.get_tags_from_note(id);
    tags.get_available();

    window.sessionStorage.setItem('id', id);

    $('#modify-tag-button').click(function() {
      var data = {};
      data.title = $('#title').val();
      data.description = $('#description').val();

      if($('#title').val() != '' && $('#description').val() != '')
        notes.patch(data, window.sessionStorage.getItem('id'));
    })
  },

```

```
setData: function(data) {
  document.getElementById('header').innerHTML = data.title;
  document.getElementById('title').value= data.title;
  document.getElementById('description').value= data.description;
},

```

```
buildTable: function(data) {
  var table = $('#tabla-etiquetas').DataTable({
    data: data,
    columns: [
      { title: "Título ", data: "tag_name"},
      { title: "Descripción", data: "description"},
      { title: "Fecha de creación", data: "created_at"},
      { title: "Detalles", data : null, "defaultContent": "<div class='text-center'><button"},
      { title: "Eliminar", data : null, "defaultContent": "<div class='text-center'><button"}
    ]
  });

  $('#tabla-etiquetas tbody').on('click', '.btn-success', function() {
    var data = table.row($(this).parents('tr')).data();
    window.location.href = "notes_from_tag.html?id=" + data.id;
  });
}
```




MENU ETIQUETAS – VISTA

- ▶ Además de las notas, también podemos mostrar todas las etiquetas que el usuario a creado. Además, se podrá crear una nueva etiqueta, ver el detalle de una etiqueta en particular y borrar una etiqueta que escoja el usuario.

Aplicación de Notas Grupo 2

TSC 2016-2

Bienvenido: carlos.juregui@alumnos.uan.cl Recargar vista Logout

Notas

Etiquetas

Notas

Crear Nota

Show 10 entries Search:

Titulo	Descripción	Fecha de creación	Detalles	Eliminar
Certamen	Certamen de base de datos el miércoles a las 3:40.	2016-11-16T02:38:13.270Z	Ver	
Laboratorio	Laboratorio de física a las 11:30 de la mañana.	2016-11-16T02:35:22.250Z	Ver	

Showing 1 to 2 of 2 entries

Previous 1 Next

Crear Nota

Titulo

Descripción

Crear



MENU ETIQUETAS – CÓDIGO

- ▶ Para el menú de etiquetas, mandamos una petición a la API de tipo GET, recibiendo así un información en formato *JSON*, el cual se lo pasamos a la función `buildTable`, quien nos genera la tabla con las etiquetas existentes.

```
var tagsView = {
  init: function() {
    navbarView.init();
    tags.get_all();

    $('#new-tag-button').click(function() {
      var data = {};
      data.tag_name = $('#title').val();
      data.description = $('#description').val();
      tags.post(data);
    })
  },

```

```
  buildTable: function(data) {
    var table = $('#tabla-etiquetas').DataTable({
      data: data,
      columns: [
        { title: "Título ", data: "tag_name"},
        { title: "Descripción", data: "description"},
        { title: "Fecha de creación", data: "created_at"},
        { title: "Detalles", data : null, "defaultContent": "<div class='text-center'>"
        { title: "Elininar", data : null, "defaultContent": "<div class='text-center'>"
      ]
    });
  },

```

```
var tags = {
  get_all: function() {
    $.ajax({
      url: "http://localhost:9999/tags/",
      data: {},
      type: "GET",
      dataType: "json",
      beforeSend: function(xhr) {
        token = btoa(window.localStorage.getItem('email_user') + ':' + window.localStorage.getItem('password_user'));

        xhr.setRequestHeader("Authorization", token);
      },
      success: function(json) {
        tagsView.buildTable(json);
      },
      error: function() {
        alert("Error al obtener los datos de las etiquetas.");
      },
      complete: function() {
      }
    });
  },
};

```



BORRAR ETIQUETA – VISTA

- ▶ Para borrar una etiqueta, seguimos el mismo procedimiento que las notas, haciendo click en el icono rojo con un basurero. Posteriormente, la aplicación nos preguntara si estamos seguros de eliminar la etiqueta. Si nuestra respuesta es afirmativa, la etiqueta se borrara, mostrando un mensaje de éxito.

mat4	Matematicas 4.	2016-11-16T03:01:51.820Z	Ver	
------	----------------	--------------------------	-----	--

Eliminar etiqueta

¿Seguro que desea eliminar esta etiqueta?

No Sí, eliminar

Eliminar etiqueta

¿Seguro que desea eliminar esta etiqueta?

Etiqueta borrada con éxito.

Aceptar



BORRAR ETIQUETA - VISTA

- Aquí lo que hacemos es crear una función para el botón de confirmar el borrado de una etiqueta, donde a nuestra variable *tags* le pasamos el id de la etiqueta a borrar, llamando al método `delete`. Con esto mandamos una petición a la API de tipo **DELETE** con la id de la etiqueta a borrar, borrando en la API dicha etiqueta y recargando la vista.

```
delete: function(id) {
  $.ajax({
    url: "http://localhost:9999/tags/" + id,
    contentType: "application/json; charset=UTF-8",
    dataType: "json",
    type: "DELETE",
    processData: false,
    timeout: 30000,
    beforeSend: function(xhr) {
      token = btoa(window.localStorage.getItem('email_user') + ':' + window.localStorage.getItem('password_user'));

      xhr.setRequestHeader("Authorization", token);
    },
    success: function() {
      alert("Etiqueta borrada con éxito.");
      location.reload();
    },
    error: function() {
      alert("Error al borrar la etiqueta.");
    },
    complete: function() {
    }
  });
},
catch: function(data, id) {
```

```
buildTable: function(data) {
  var table = $('#tabla-etiquetas').DataTable({
    data: data,
    columns: [
      { title: "Título ", data: "tag_name"},
      { title: "Descripción", data: "description"},
      { title: "Fecha de creación", data: "created_at"},
      { title: "Detalles", data: null, "defaultContent": "<div class='text-center'><button"},
      { title: "Eliminar", data: null, "defaultContent": "<div class='text-center'><button"}
    ]
  });

  $('#tabla-etiquetas tbody').on('click', '.btn-success', function() {
    var data = table.row($(this).parents('tr')).data();
    window.location.href = "notes_from_tag.html?id=" + data.id;
  });

  $('#tabla-etiquetas tbody').on('click', '.btn-danger', function() {
    var data = table.row($(this).parents('tr')).data();
    window.sessionStorage.setItem("selected_tag", data.id);
    $('#tag-delete-modal').modal('show');
  });

  $('#confirm-tag-delete').click(function() {
    var erased_tag = window.sessionStorage.getItem("selected_tag");
    tags.delete(erased_tag);
  });
}
```




DETALLE DE UNA ETIQUETA - VISTA

- En el detalle de una etiqueta, podremos ver la información de una etiqueta, y además podremos modificarla (los campos se auto-rellenarán con los datos de la etiqueta). Por otra parte, podremos ver las notas asignadas a esta etiqueta.

Modificar etiqueta

Título

fis4

Descripción

Física 140.

Modificar

fis4

Modificar etiqueta

Notas

Show 10 entries Search:

Título	Descripción	Fecha de creación	Detalles	Eliminar
Laboratorio	Laboratorio de física a las 11:30 de la mañana.	2016-11-16T02:35:22.250Z	Ver	

Showing 1 to 1 of 1 entries Previous 1 Next



DETALLE DE UNA ETIQUETA - CÓDIGO

- De forma similar a las notas, aquí guardamos las variables el nombre y descripción de la etiqueta, para posteriormente mostrarlas con la función `getElementById`. También desplegamos todas las notas asociadas a esta etiqueta.

```
buildTable: function(data) {  
  var table = $('#tabla-notes').DataTable({  
    data: data,  
    columns: [  
      { title: "Título ", data: "title"},  
      { title: "Descripción", data: "description"},  
      { title: "Fecha de creación", data: "created_at"},  
      { title: "Detalles", data: null, "defaultContent": "<div class='text-center'><button  
      { title: "Eliminar", data: null, "defaultContent": "<div class='text-center'><button  
    ]  
  });  
}
```

```
$('#tabla-notes tbody').on('click', '.btn-success', function() {  
  var data = table.row($(this).parents('tr')).data();  
  window.location.href = "tags_from_note.html?id=" + data.id;  
});  
  
$('#tabla-notes tbody').on('click', '.btn-danger', function() {  
  var data = table.row($(this).parents('tr')).data();  
  window.sessionStorage.setItem("selected_note", data.id);  
  $('#note-delete-modal').modal('show');  
});
```

```
var notesFromTagView = {  
  init: function(id) {  
    navbarView.init();  
    tags.get_unique(id);  
    notes.get_notes_from_tag(id);  
  
    window.sessionStorage.setItem('id', id);  
  
    $('#modify-note-button').click(function() {  
      var data = {};  
      data.tag_name = $('#title').val();  
      data.description = $('#description').val();  
  
      if($('#title').val() != '' && $('#description').val() != '')  
        tags.patch(data, window.sessionStorage.getItem('id'));  
      else  
        alert("Debe rellenar todos los campos.");  
    })  
  },  
}
```

```
setData: function(data) {  
  document.getElementById('header').innerHTML = data.tag_name;  
  document.getElementById('title').value = data.tag_name;  
  document.getElementById('description').value = data.description;  
},
```



ASIGNAR UNA ETIQUETA A UNA NOTA – VISTA

- Aquí podremos ver las etiquetas disponibles para asignar. Solo debemos presionar el botón con un icono de etiqueta para seleccionar la nota que asignaremos. Finalmente, si todo sale bien, se mostrara un mensaje de éxito en la aplicación.

Etiquetas Disponibles

Show 10 entries

Search:

Título	Descripción	Asignar
fis4	Fisica 140.	

Showing 1 to 1 of 1 entries

Previous 1 Next

Etiqueta asignada con éxito.

Aceptar



ASIGNAR UNA ETIQUETA A UNA NOTA – CÓDIGO

- Aquí guardamos la nota que será asignada, y la etiqueta seleccionada para asignar. Luego, llamamos a la función `assign_tag`, la cual mandara una petición de tipo POST a la API, con la id de la nota en la URL, y mandando el id de la etiqueta. Si todo sale bien, se mandara un mensaje de éxito, y se recargara la vista.

```
assign_tag: function(note_id, tag_id) {
  $.ajax({
    url: "http://localhost:9999/notes/" + note_id + "/assign_tag/",
    data: JSON.stringify({
      tag_id: tag_id
    }),
    contentType: "application/json; charset=UTF-8",
    dataType: "json",
    type: "POST",
    processData: false,
    timeout: 30000,
    beforeSend: function(xhr) {
      token = btoa(window.localStorage.getItem('email_user') + ':' + window.localStorage.getItem('password_user'));

      xhr.setRequestHeader("Authorization", token);
    },
    success: function() {
      alert("Etiqueta asignada con éxito.")
      location.reload();
    },
    error: function() {
      alert("Error al asignar la etiqueta.")
    },
    complete: function() {
    }
  });
},
```

```
buildTableAvailables: function(data) {
  var table = $('#tabla-etiquetas-disponibles').DataTable({
    data: data,
    columns: [
      { title: "Título", data: "tag_name"},
      { title: "Descripción", data: "description"},
      { title: "Asignar", data : null, "defaultContent": "<div class='text-center'><button"
    ]
  });

  $('#tabla-etiquetas-disponibles tbody').on('click', '.btn-primary', function() {
    var data = table.row($(this).parents('tr')).data();
    notes.assign_tag(window.sessionStorage.getItem('id'), data.id);
  });
}
```


AUTENTICACIÓN OAUTH



TSC 2016-2



CAMBIOS EN LA API

► holi

TSC 2016-2



CAMBIOS EN NWJS

► holi2

FIN