

RELOJ

El actual gerente de la empresa desea que se implemente un mecanismo de comunicación interna que permite enviar mensajes de una persona a otra registrando la hora.

Esta hora siempre es enviada por el servidor ya que es de donde predominan las peticiones.

SOLUCION

Se implementara la siguiente solución:



Comunicacion Interna

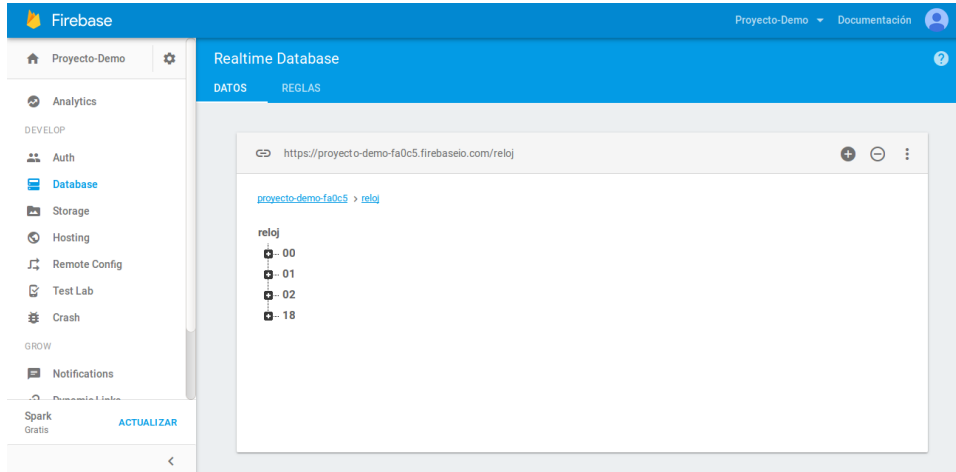
Cada persona escribirá el nombre de origen y el nombre de destino, en conjunto con el mensaje y este mensaje será guardado en la base de datos y podrá ser reflejado en el esquema de la siguiente manera



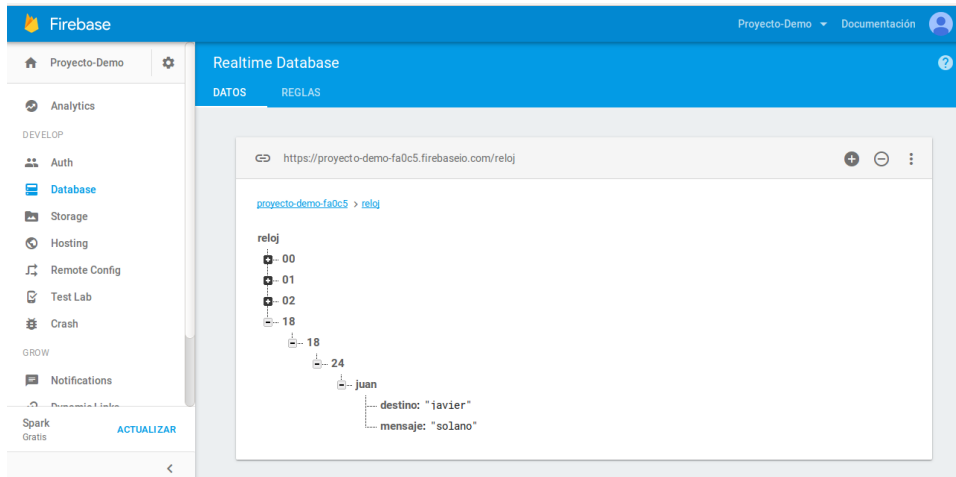
IMPLEMENTACION:

Descargar el Código de Ejemplo 03 y realizar el mismo procedimiento al Código de Ejemplo 02

Crear el esquema de reloj en la base de datos de la aplicación en el firebase y a su vez importar el archivo database.json a reloj y la estructura final será la siguiente.



Insertar el esquema de base de datos, la cual se lee a las 18:18:24 Juan le envió un mensaje a Javier con el contenido de solano



EXPLICACIÓN CÓDIGO

Se desea guardar la información como la hora, el origen, el destinatario y el mensaje, para ello necesitaremos de lo siguiente:

- Retornar la hora(hora, minuto y segundo)
- Retornar información del evento(origen, destinatario y mensaje)
- Conexión a la base de datos
- Evento que estará esperando una acción para ejecutar el guardado

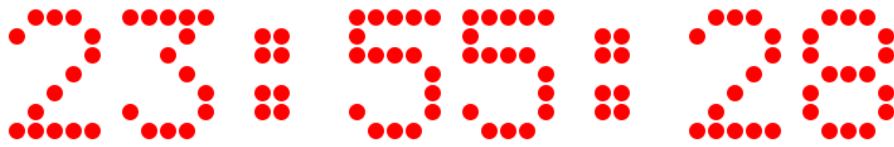
Lo primero que haremos será agregar al archivo las credenciales de nuestro proyecto, sin estas credenciales no podremos acceder a nuestro proyecto.

```
var config = {  
  apiKey: "AIzaSyDtxS_3-dszy3_jjz1JCr8v9xtbwBIpdSs",  
  authDomain: "proyecto-demo-fa0c5.firebaseio.com",  
  databaseURL: "https://proyecto-demo-fa0c5.firebaseio.com",  
  storageBucket: "proyecto-demo-fa0c5.appspot.com",  
};  
firebase.initializeApp(config);
```

Posteriormente comenzaremos a correr el siguiente comando en el Git Bash y nos dirigimos a la url que nos provee

```
→ reloj_hackspace firebase serve  
Starting Firebase development server...  
  
Project Directory: /home/carlos/reloj_hackspace  
Public Directory: ./  
  
Server listening at: http://localhost:5000
```

Al entrar a la url propuesta, en este caso <http://localhost:5000> , la vista que debería de ser mostrada debería ser la siguiente:



Comunicacion Interna

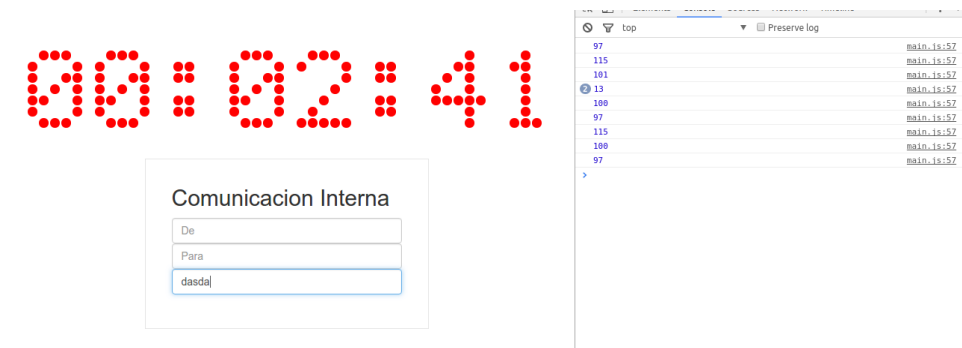
Comenzaremos escribiendo el código correspondiente al evento que estará escuchando cuando presionemos una tecla mientras estemos escribiendo el mensaje, el cual ejecutara un guardado den la base de datos de firebase de la aplicación cuando la tecla presionada sea ENTER.

```
$(document).ready(function(){  
  $('#mensaje').on('keypress',function(e){  
    console.log(e.which);|  
  });  
});
```

En la imagen se aprecia el código que realiza lo siguiente:

- Espera a que todo el HTML sea cargado
- Selecciona todos los elementos del DOM con id=mensaje se les agrega un evento cuando la tecla sea presionada (keypress).
- La función imprimirá el código de la tecla presionada

A su vez si nos dirigimos al navegador abrimos el debugger(presionando la tecla F12), y seleccionamos el elemento que tiene como placeholder Mensaje y comenzamos a presionar las teclas podremos ver el id de cada una, y gracias a ello podremos ver que el id del ENTER es el 13.



Posteriormente modificaremos el código para enviar los datos del evento a guardar (destinatario, origen y mensaje).

En este caso modificamos la función creada previamente y le haremos una validación de que si el código de la tecla presionada es 13(en alusión al ENTER) realizar la acción que corresponde a guardar el mensaje y para ello le pasamos los valores, a su vez creamos las funciones que nos retornan estos valores realizando una selección por id.

```
function getOrigin(){
  return $('#origen').val();
}

function getDestiny(){
  return $('#destino').val();
}

function getMessage(){
  return $('#mensaje').val();
}

function storeMessage(origin, destiny, message){
  console.log(origin+"-"+destiny+"-"+message);
}

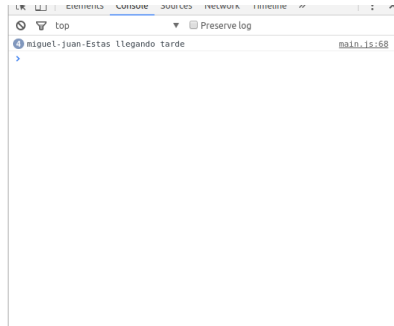
$(document).ready(function(){
  $('#mensaje').on('keypress', function(e){
    if(e.which==13){
      storeMessage(getOrigin(),getDestiny(),getMessage());
    }
  });
});
```

Si regresamos al navegador y presionamos F5 para actualizar el main.js que mantiene la aplicación y digitamos un origen y un destinatario y posteriormente un mensaje y finalizamos presionando un ENTER, se nos imprimirá esos valores en la consola

00:09:53

Comunicacion Interna

miguel
juan
Estas llegando tarde



Ahora necesitaremos obtener el tiempo actual (hora, minutos y segundos).

En el código que podemos apreciar se realizan las siguientes acciones, como cada tiempo actual (hora, minuto y segundo) cuenta con 2 dígitos uno izquierdo y uno derecho se debe de juntar ambos números para obtener el valor real, a su vez cuando retornemos el elemento solamente queremos el ultimo dígito y por esa razón realizamos un `slice(-1)`.

```
function getHour(){
  return hh.className.slice(-1)+hx.className.slice(-1);
}

function getMinute(){
  return mm.className.slice(-1)+mx.className.slice(-1);
}

function getSecond(){
  return ss.className.slice(-1)+sx.className.slice(-1);
}

function storeMessage(origin, destiny, message){
  console.log(getHour()+"-"+getMinute()+"-"+getSecond());
  console.log(origin+"-"+destiny+"-"+message);
}
```

Si guardamos y recargamos el navegador y volvemos a escribir el origen, destinatario y el mensaje y en la consola de debug nos debe de mostrar mensajes como el siguiente:

- La hora del evento registrado
- La información del evento

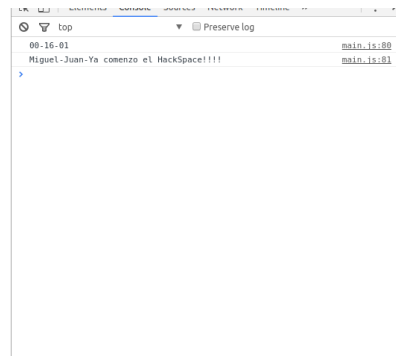
00:16:16

Comunicacion Interna

Miguel

Juan

Ya comenzo el HackSpace!!!!



Lo que haremos a continuación será almacenar esa información en firebase

Modificaremos la función storeMessage para que suba la data obtenida, lo que se puede apreciar es como se instancia la base de datos se le asigna un ref que es la referencia de la ruta, como se desea simular la estructura antes presentada en la figura 02, se juntan: la hora, el minuto, el segundo y el origen todos separados por un '/' y posteriormente se referencia a la función set y se pasan los campos a agregar en forma de llave valor, que significa que se agregara un campo llamado mensaje con la data que tiene la variable message

```
function storeMessage(origin,destiny,message){  
  firebase.database().ref(  
    'reloj/'+getHour()+ '/' +getMinute()+ '/' +getSecond()+ '/' +origin  
  ).set({  
    mensaje : message,  
    destino : destiny  
  });  
}
```

Si volvemos a refrescar el navegador y escribimos todo de manera normal

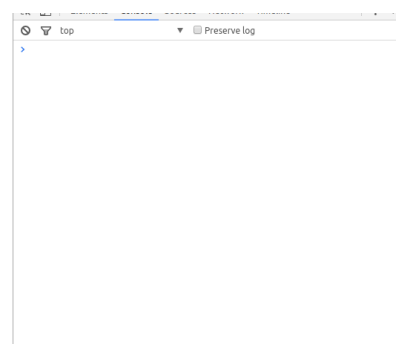
00:22:30

Comunicacion Interna

Maria

Juana

Primer mensaje subido!!!!



Y si nos dirigimos a la consola de administración de nuestra aplicación en el firebase podremos ver ese registro agregado



CONCLUSIONES:

Al finalizar este pequeño tutorial, se podrá entender el flujo de guardado de información en la base de datos con firebase