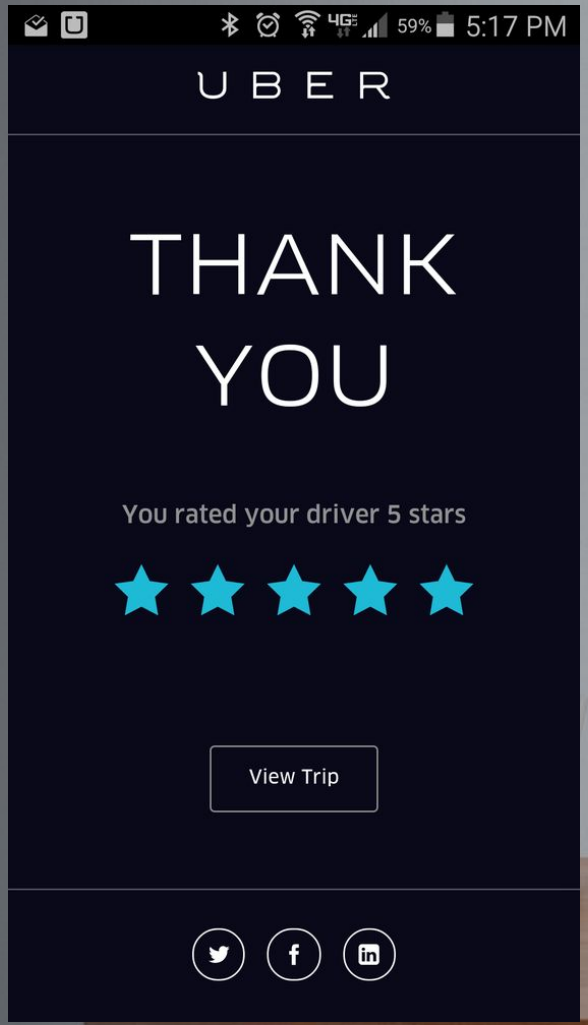

Black Mirror

Índice

¿Qué es?

Sistema de calificación del comportamiento de las personas.



¿Quiénes lo usan?

¿Qué podemos calificar?

Cualquier tipo de interacción
entre personas.

Aplicaciones:

Servicios:

Salud:

1. Calificar la atención de médicos y enfermeros

Aplicaciones:

Servicios:

Educación:

1. Calificar la enseñanza de maestros en escuelas, colegios, universidades ,etc.

Aplicaciones:

Servicios:

Calificación de cualquier tipo de servicio recibido:

- Delivery
- Transporte
- Etc

Problema Solucionado

1. Satisfacción del usuario consumidor.
2. Satisfacción del usuario que brinda el servicio.
3. Mejores auditorias

Aplicaciones:

Ranking de personas:

Influyentes (4 - 5 estrellas)

Su comportamiento es aceptado socialmente y se consideran ciudadanos ejemplares, por lo que logran obtener mayores beneficios.

Aplicaciones:

Ranking de personas:

Promedio (2.5 - 3.9 estrellas)

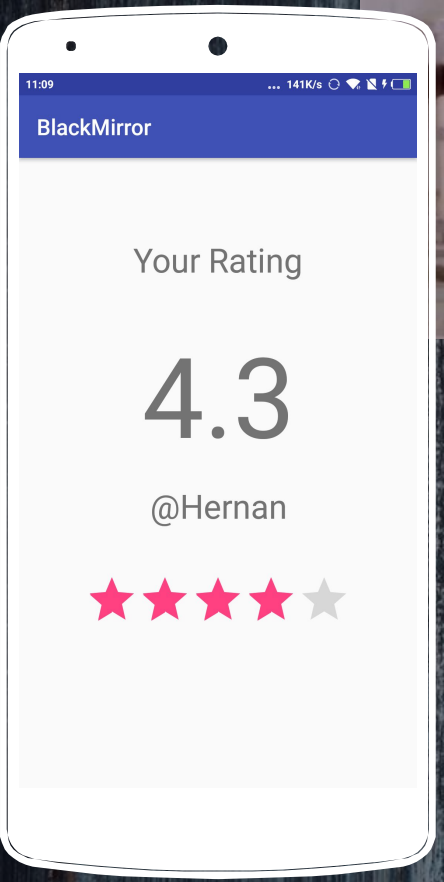
Su comportamiento normal, con altos y bajos; este grupo asociará a la mayoría de personas.

Aplicaciones:

Ranking de personas:

No deseables (1 - 2.4 estrellas)

Su comportamiento no es aceptado socialmente y se consideran malos ciudadanos y/o personas no deseables.

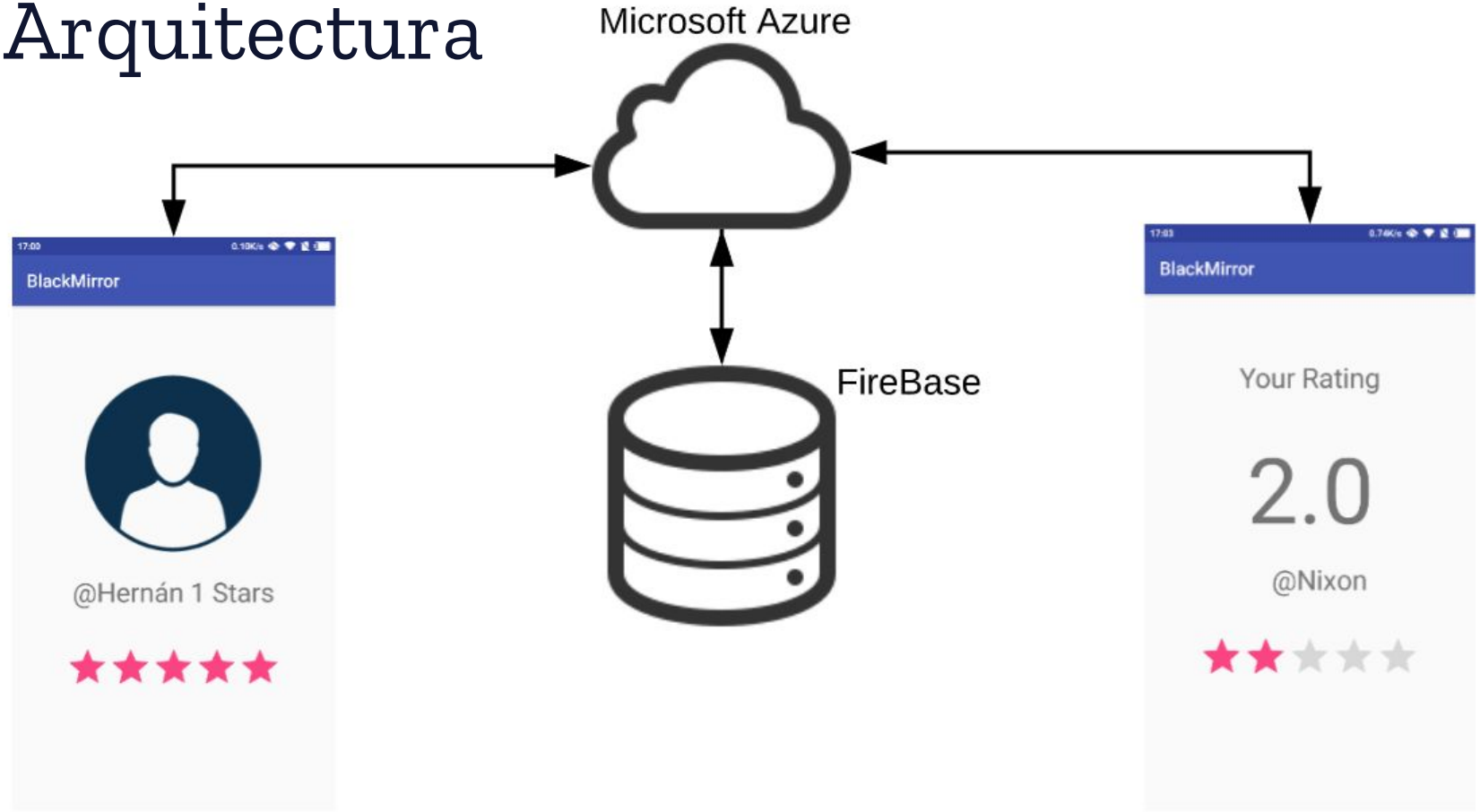

11:09
141K/s
BlackMirror
Your Rating
4.3
@Hernan


Jack
3.7 97

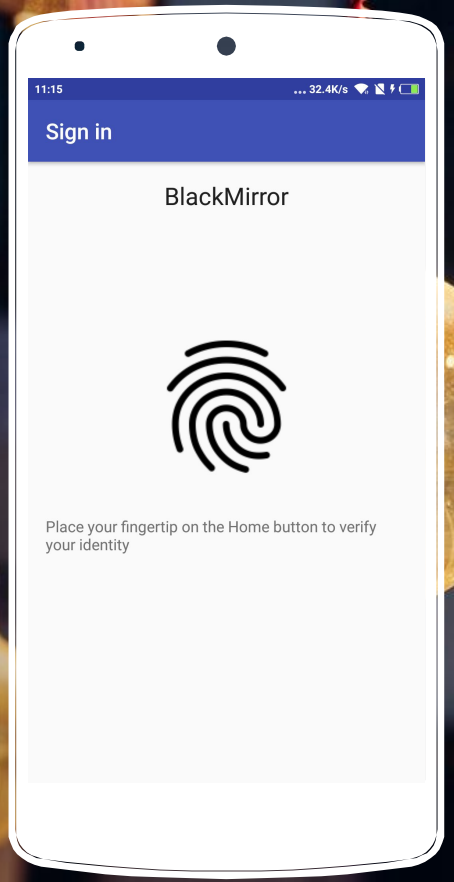
Visión


Lacie
4.2 43

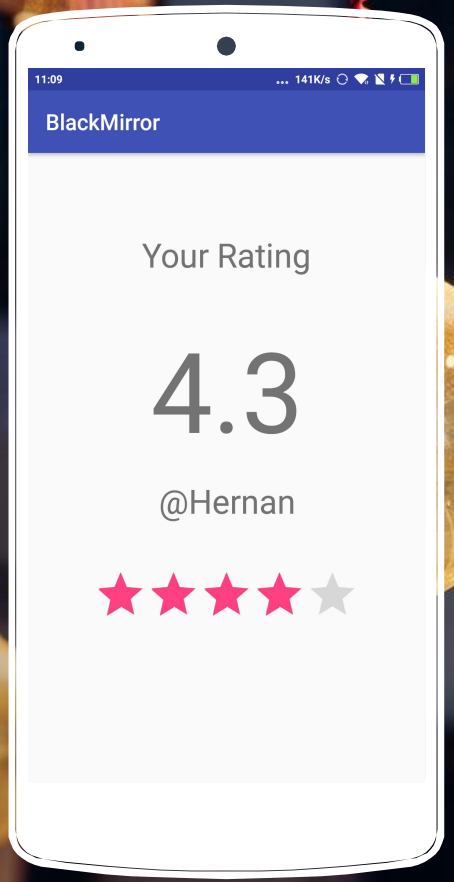
Arquitectura



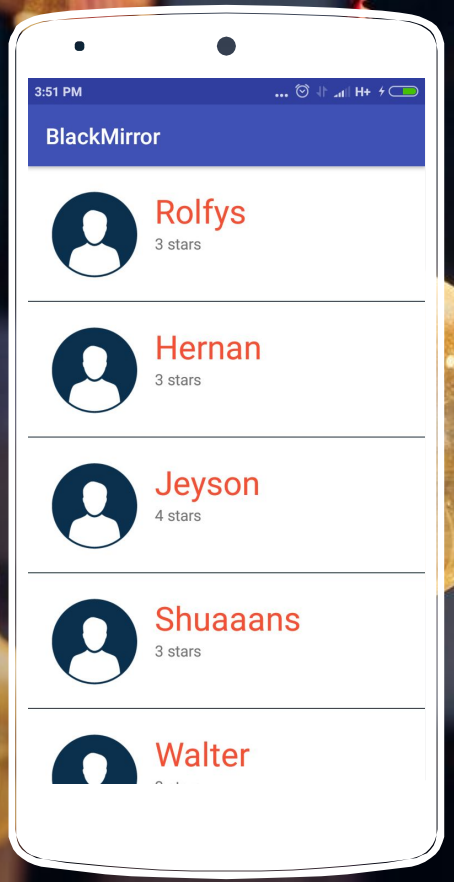
Funcionamiento



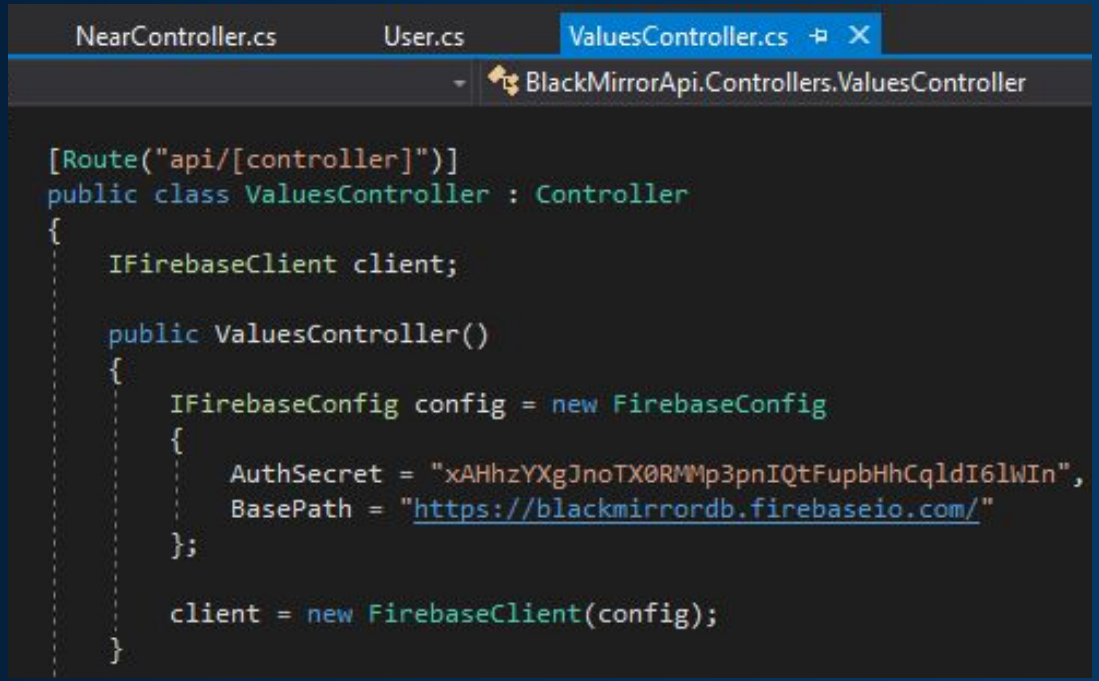
Funcionamiento

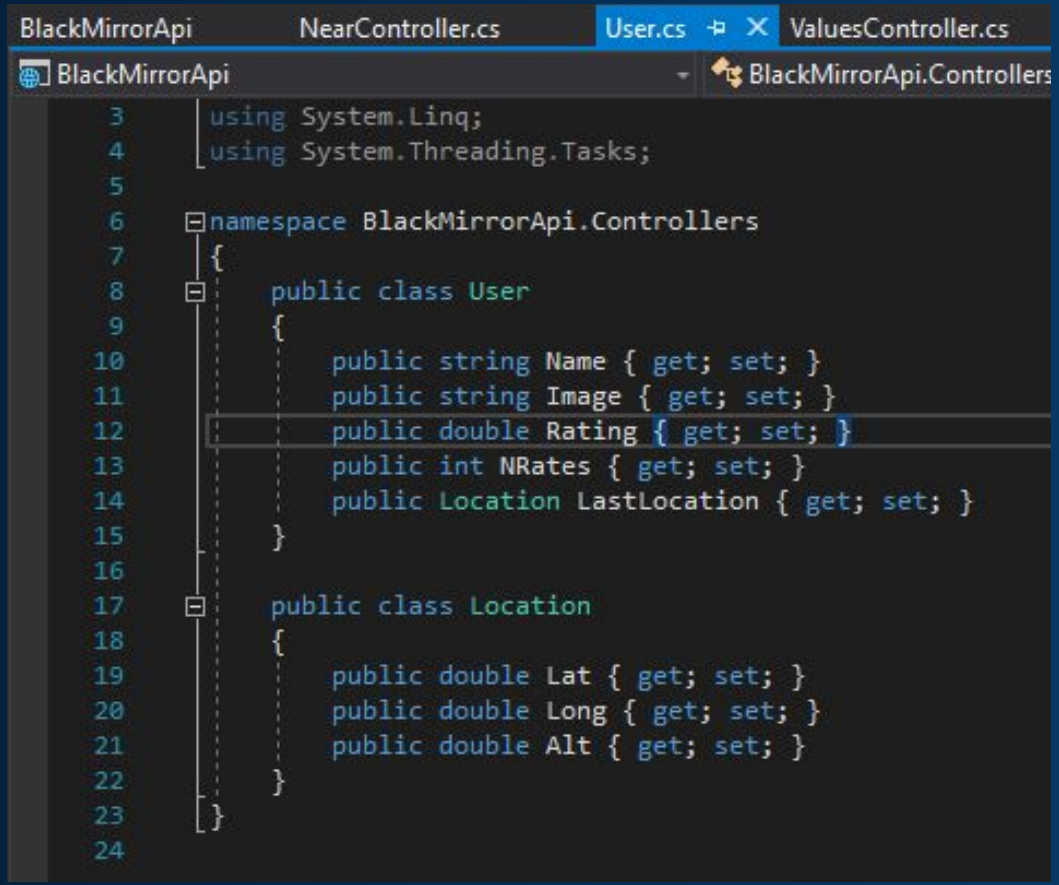


Funcionamiento



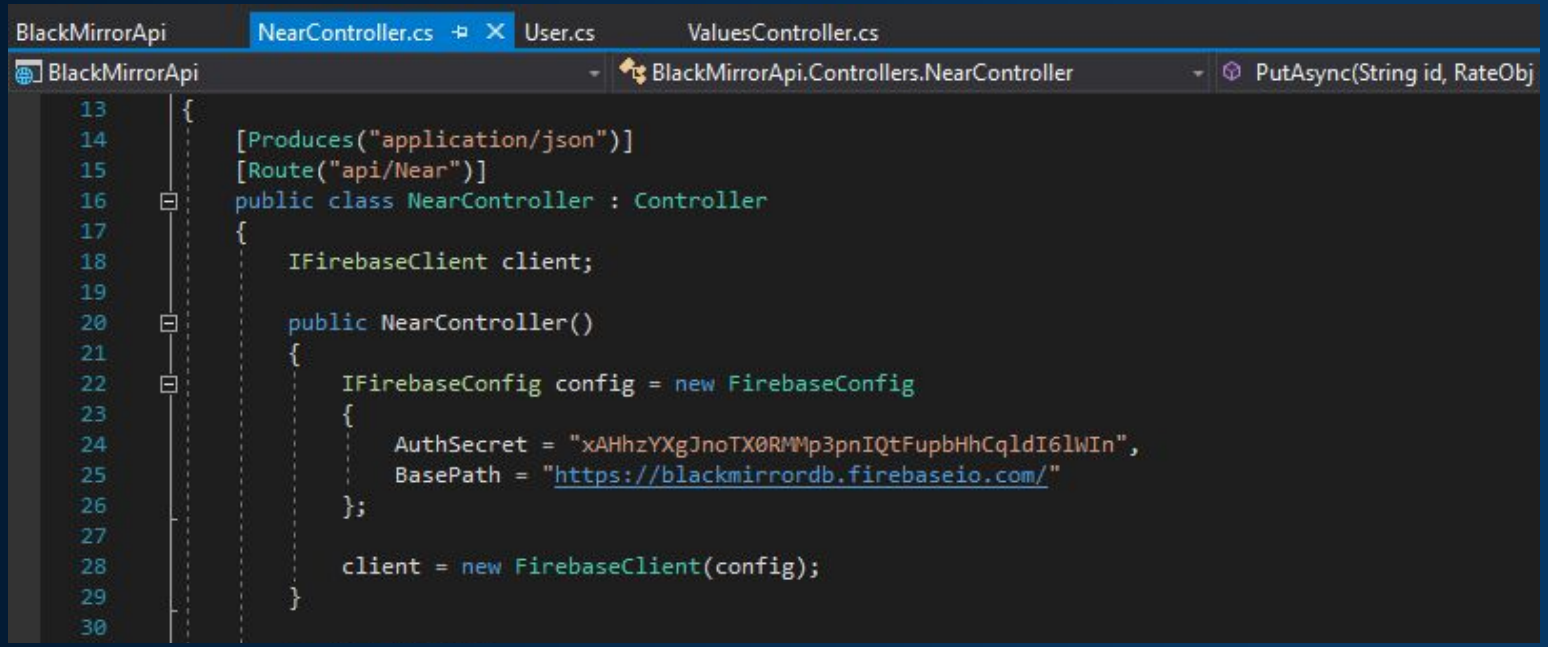
Manejo del Usuario




BlackMirrorApi
NearController.cs
User.cs
ValuesController.cs
BlackMirrorApi
BlackMirrorApi.Controllers
using System.Linq;
using System.Threading.Tasks;
namespace BlackMirrorApi.Controllers
{
public class User
{
public string Name { get; set; }
public string Image { get; set; }
public double Rating { get; set; }
public int NRates { get; set; }
public Location LastLocation { get; set; }
}
public class Location
{
public double Lat { get; set; }
public double Long { get; set; }
public double Alt { get; set; }
}
}

```
// GET api/values
[HttpGet]
public async Task<IDictionary<string, User>> GetObjectsAsync()
{
    FirebaseResponse response = await client.GetAsync("user/");
    IDictionary<string, User> users = response.ResultAs<IDictionary<string, User>>();
    return users;
}

// GET api/values/5
[HttpGet("{id}")]
public async Task<User> GetAsync(string id)
{
    FirebaseResponse response = await client.GetAsync("user/" + id);
    return response.ResultAs<User>();
}

// POST api/values
[HttpPost]
public async Task<Key> PostAsync([FromBody]User user)
{
    PushResponse response = await client.PushAsync("user/", user);
    return new Key(response.Result.name);
}

// PUT api/values/5
[HttpPut("{id}")]
public async Task<User> PutAsync(String id, [FromBody]User user)
{
    FirebaseResponse response = await client.UpdateAsync("user/" + id, user);
    return response.ResultAs<User>();
}
```

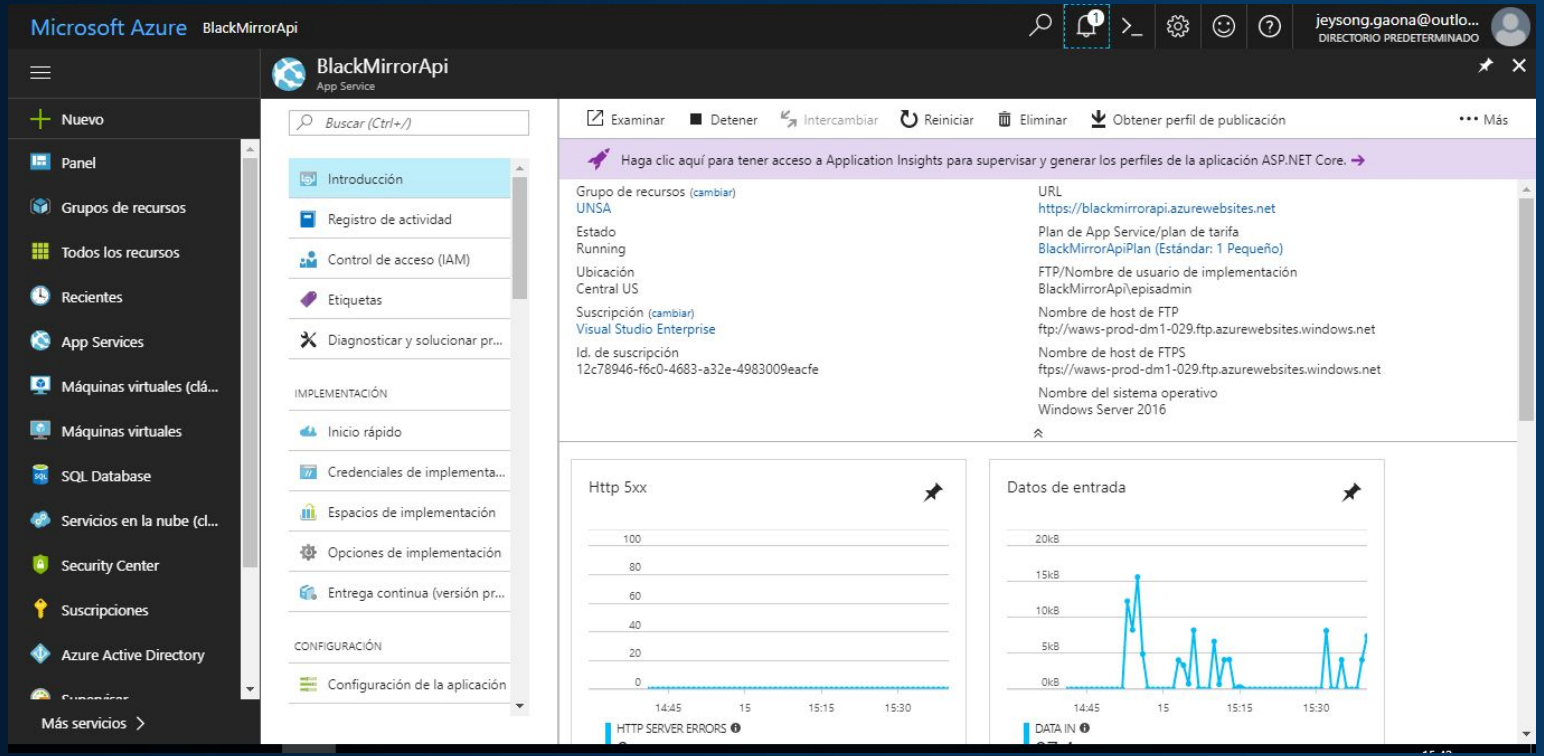
Obtener usuarios cercanos



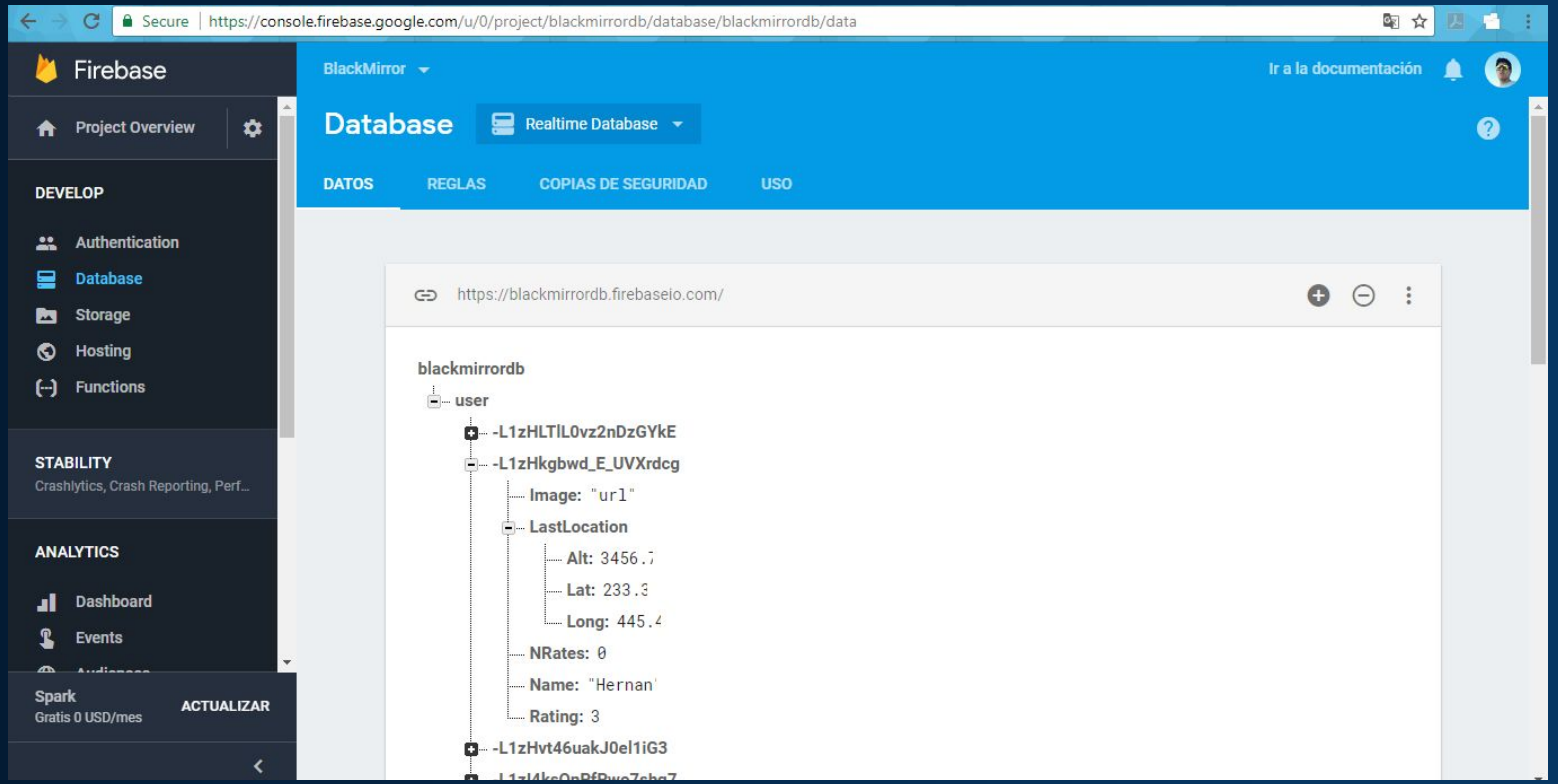
```
public static double DistanceTo(Location loc1, Location loc2)
{
    double lat1 = loc1.Lat;
    double lon1 = loc1.Long;
    double lat2 = loc2.Lat;
    double lon2 = loc2.Long;

    double rlat1 = Math.PI * lat1 / 180;
    double rlat2 = Math.PI * lat2 / 180;
    double theta = lon1 - lon2;
    double rtheta = Math.PI * theta / 180;
    double dist =
        Math.Sin(rlat1) * Math.Sin(rlat2) + Math.Cos(rlat1) *
        Math.Cos(rlat2) * Math.Cos(rtheta);
    dist = Math.Acos(dist);
    dist = dist * 180 / Math.PI;
    dist = dist * 60 * 1.1515;

    return dist * 1.609344 * 1000;
}
```

```
// PUT api/near/#
[HttpPut("{id}")]
public async Task<User> PutAsync(String id, [FromBody]RateObj rate)
{

    FirebaseResponse response = await client.GetAsync("user/" + id);
    User user = response.ResultAs<User>();

    user.Rating = Math.Round((user.Rating * user.NRates + rate.Rate) / (user.NRates + 1.0), 1);
    user.NRates = user.NRates + 1;

    FirebaseResponse response2 = await client.UpdateAsync("user/" + id, user);
    return response2.ResultAs<User>();
}
```

API en AZURE


Microsoft Azure
BlackMirrorApi
jeysong.gaona@outlo...
DIRECTORIO PREDETERMINADO
BlackMirrorApi
App Service
Nuevo
Panel
Grupos de recursos
Todos los recursos
Recientes
App Services
Máquinas virtuales (clá...
Máquinas virtuales
SQL Database
Servicios en la nube (cl...
Security Center
Suscripciones
Azure Active Directory
Más servicios
Buscar (Ctrl+/)
Introducción
Registro de actividad
Control de acceso (IAM)
Etiquetas
Diagnosticar y solucionar pr...
IMPLEMENTACIÓN
Inicio rápido
Credenciales de implementa...
Espacios de implementación
Opciones de implementación
Entrega continua (versión pr...
CONFIGURACIÓN
Configuración de la aplicación
Examinar
Detener
Intercambiar
Reiniciar
Eliminar
Obtener perfil de publicación
Más
Haga clic aquí para tener acceso a Application Insights para supervisar y generar los perfiles de la aplicación ASP.NET Core.
Grupo de recursos (cambiar)
UNSA
Estado
Running
Ubicación
Central US
Suscripción (cambiar)
Visual Studio Enterprise
Id. de suscripción
12c78946-f6c0-4683-a32e-4983009eacfe
URL
https://blackmirrorapi.azurewebsites.net
Plan de App Service/plan de tarifa
BlackMirrorApiPlan (Estándar: 1 Pequeño)
FTP/Nombre de usuario de implementación
BlackMirrorApi\episadmin
Nombre de host de FTP
ftp://waws-prod-dm1-029.ftp.azurewebsites.windows.net
Nombre de host de FTPS
ftps://waws-prod-dm1-029.ftp.azurewebsites.windows.net
Nombre del sistema operativo
Windows Server 2016
Http 5xx
100
80
60
40
20
0
14:45
15
15:15
15:30
HTTP SERVER ERRORS
Datos de entrada
20kB
15kB
10kB
5kB
0kB
14:45
15
15:15
15:30
DATA IN

Base de Datos en Firebase



Conclusiones

Black Mirror es una aplicación que está enfocada a:

1. Smart Education
2. Smart Health
3. Smart Governance

Referencias

1. Peis, E., Morales del Castillo, J. M., & Delgado López, J. A. (2008). Sistemas de Recomendación Semánticos. Un análisis del estado de la cuestión.
2. VARGAS, Washington Adrián Velásquez. Algoritmo de recomendación sensible a contexto de elementos educativos reutilizables con almacenamiento Orientado a Grafos. Revista Tecnológica-ESPOL, 2017, vol. 30, no 1.


Gracias!