

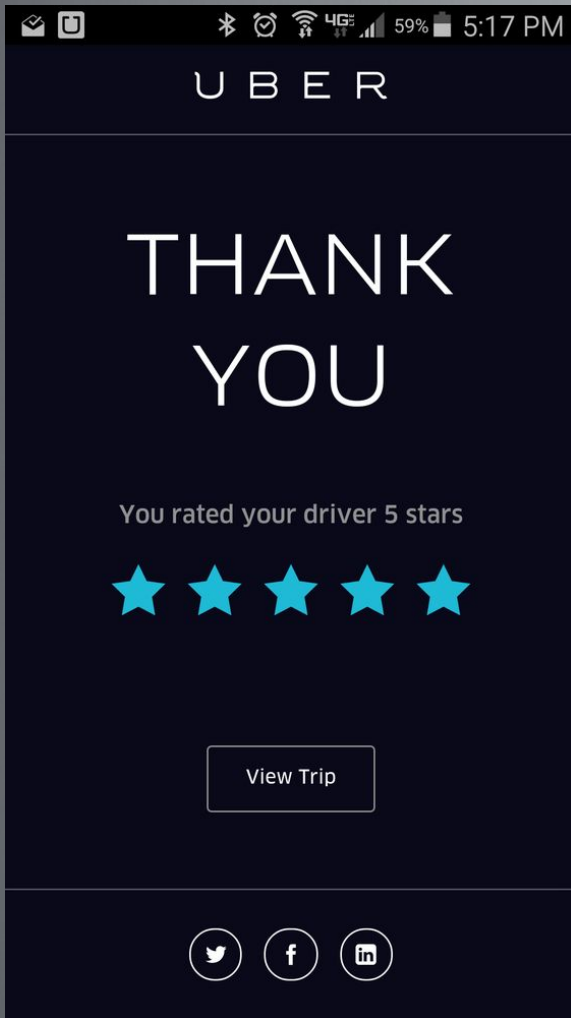
A high-angle, top-down photograph of a person standing on a white arrow painted on a dark asphalt surface. The person is wearing blue jeans and brown shoes. The arrow points downwards, towards the bottom of the frame. The text "Black Mirror" is overlaid in the bottom left corner.

Black Mirror



Qué es?

Sistema de calificación del comportamiento de las personas.



Quiénes lo usan?



A close-up photograph of a person's hand, palm up, holding a tangled string of warm white LED lights. The lights are glowing, and the background is a soft, out-of-focus blue sky. The hand is positioned in the lower half of the frame, with the fingers slightly curled around the string of lights.

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



A person with long hair, wearing a light green jacket, black pants, and white sneakers, is running towards the right. They are in front of a large, vibrant mural that depicts an underwater scene. The mural features a large hand reaching down from the top left, several fish of different sizes, and a small boat on the right. The overall color palette of the mural is dominated by shades of blue and green. The person is running on a light-colored, possibly concrete, surface.

Problema

Solucionado

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

A close-up, slightly blurred photograph of a woman's face, focusing on her wide smile and showing her teeth. Her hair is dark and curly. The background is dark and out of focus.

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)

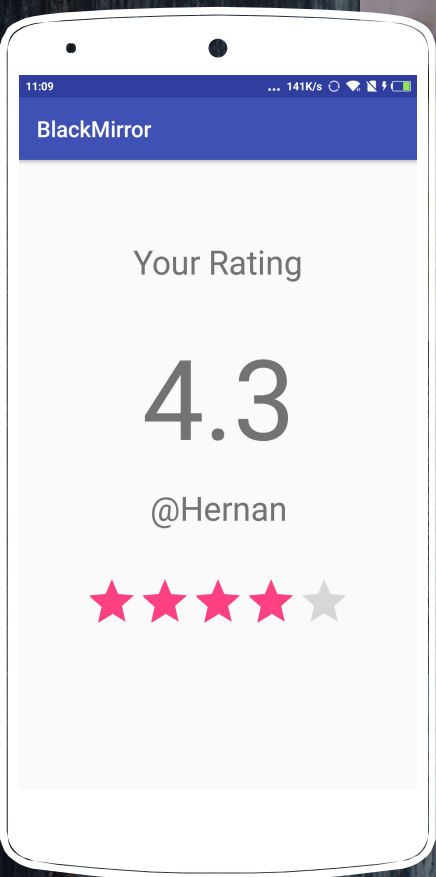
Su comportamiento normal, con altos y bajos; este grupo asociará a la mayoría de personas.

Aplicaciones:

Ranking de personas:

No deseables (1 - 2.4 estrellas)

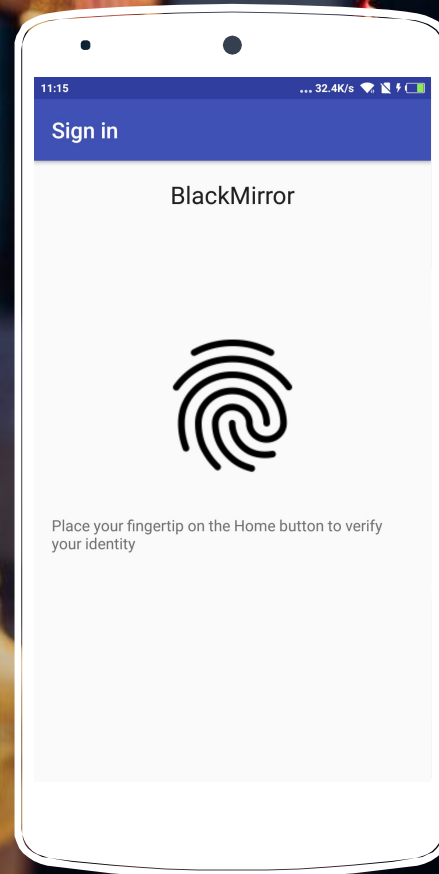
Su comportamiento no es aceptado socialmente y se consideran malos ciudadanos y/o personas no deseables.



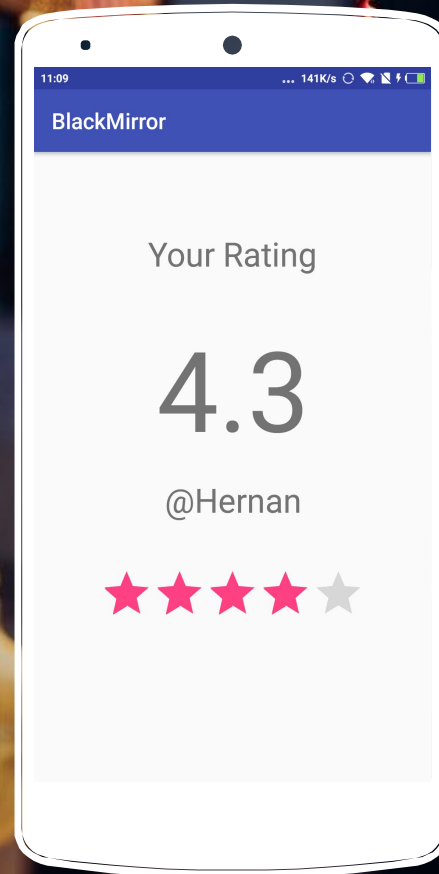
Visión



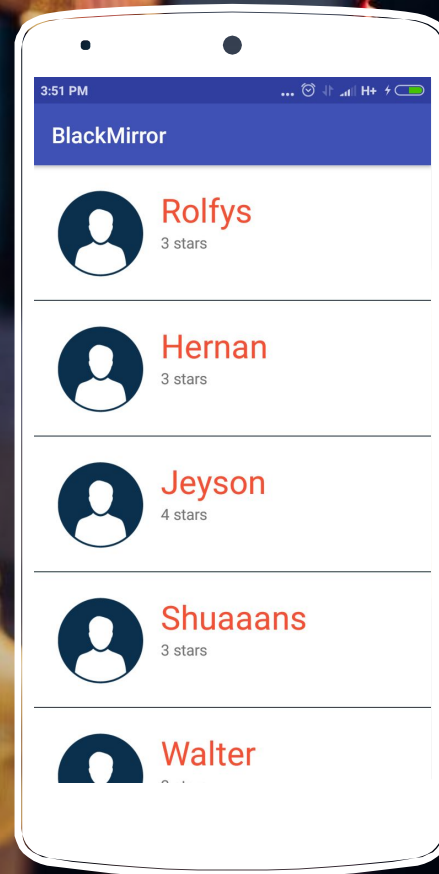
Funcionamiento



Funcionamiento



Funcionamiento



Manejo del Usuario

```
NearController.cs  User.cs  ValuesController.cs  ↗ ✕
- BlackMirrorApi.Controllers.ValuesController

[Route("api/[controller]")]
public class ValuesController : Controller
{
    IFirebaseClient client;

    public ValuesController()
    {
        IFirebaseConfig config = new FirebaseConfig
        {
            AuthSecret = "xAHhzYXgJnoTX0RMmp3pnIQtFupbHhCqldI6lWIn",
            BasePath = "https://blackmirrordb.firebaseio.com/"
        };

        client = new FirebaseClient(config);
    }
}
```

BlackMirrorApi

NearController.cs

User.cs

ValuesController.cs

BlackMirrorApi

BlackMirrorApi.Controllers

```
3 using System.Linq;
4 using System.Threading.Tasks;
5
6 namespace BlackMirrorApi.Controllers
7 {
8     public class User
9     {
10         public string Name { get; set; }
11         public string Image { get; set; }
12         public double Rating { get; set; }
13         public int NRates { get; set; }
14         public Location LastLocation { get; set; }
15     }
16
17     public class Location
18     {
19         public double Lat { get; set; }
20         public double Long { get; set; }
21         public double Alt { get; set; }
22     }
23 }
24
```



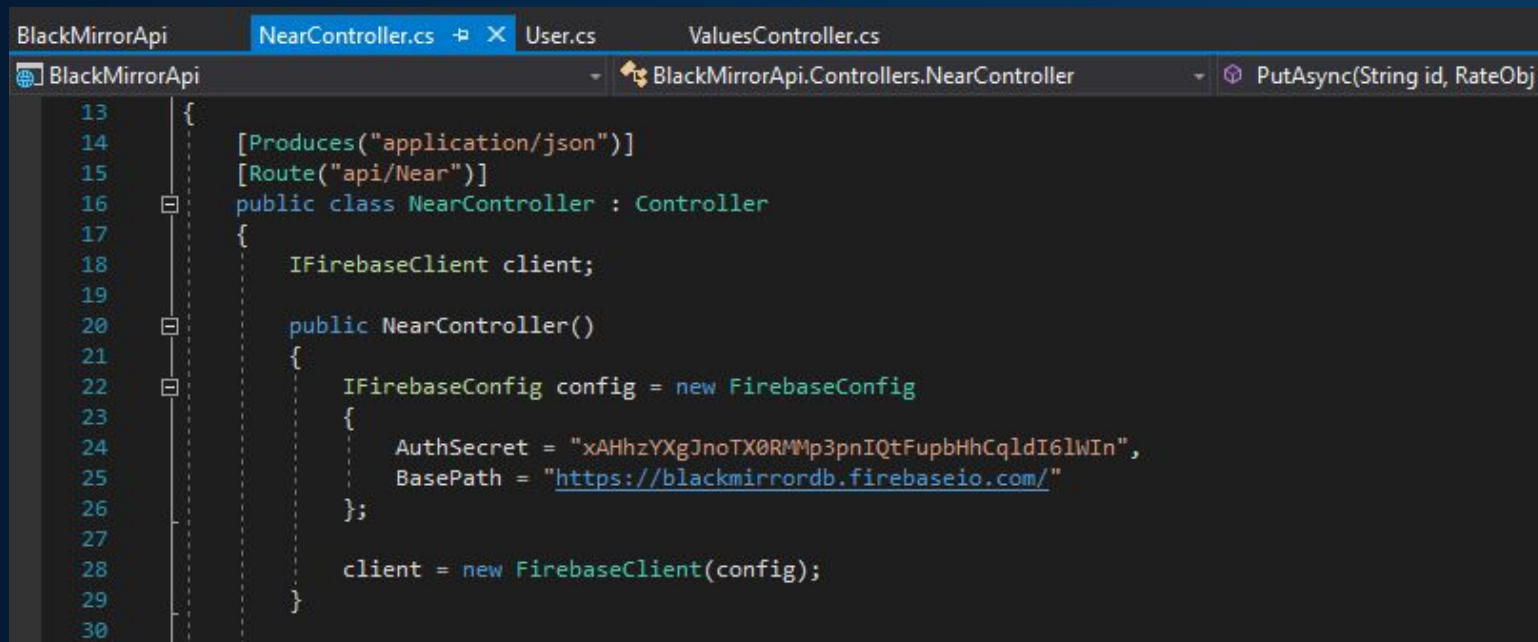
```
// GET api/values
[HttpGet]
public async Task<IDictionary<string, User>> GetObjectsAsync()
{
    FirebaseResponse response = await client.GetAsync("user/");
    IDictionary<string, User> users = response.ResultAs<IDictionary<string, User>>();
    return users;
}

// 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



```
BlackMirrorApi | NearController.cs | User.cs | ValuesController.cs
BlackMirrorApi | BlackMirrorApi.Controllers.NearController | PutAsync(String id, RateObj

13 {
14     [Produces("application/json")]
15     [Route("api/Near")]
16     public class NearController : Controller
17     {
18         IFirebaseClient client;
19
20         public NearController()
21         {
22             IFirebaseConfig config = new FirebaseConfig
23             {
24                 AuthSecret = "xAHhzYXgJnoTX0RMmp3pnIQtfupbHhCqldI6lWIn",
25                 BasePath = "https://blackmirrordb.firebaseio.com/"
26             };
27
28             client = new FirebaseClient(config);
29         }
30     }
```

```
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

The screenshot displays the Microsoft Azure portal interface for an API named 'BlackMirrorApi'. The left sidebar contains navigation options such as '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', and 'Más servicios >'. The main content area is divided into several sections:

- Top Bar:** Displays 'Microsoft Azure BlackMirrorApi' and a search icon. A notification icon with a red dot is highlighted with a dashed blue box.
- Left Panel:** Contains a search bar 'Buscar (Ctrl+ /)' and a list of navigation items: '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', and 'Configuración de la aplicación'.
- Top Action Bar:** Includes buttons for 'Examinar', 'Detener', 'Intercambiar', 'Reiniciar', 'Eliminar', and 'Obtener perfil de publicación', along with a 'Más' menu.
- Information Section:** Provides details about the API, including the URL 'https://blackmirrorapi.azurewebsites.net', the App Service plan 'BlackMirrorApiPlan (Estándar: 1 Pequeño)', the FTP user 'BlackMirrorApi\episadmin', and the operating system 'Windows Server 2016'.
- Monitoring Section:** Features two charts: 'Http 5xx' (showing HTTP server errors) and 'Datos de entrada' (showing data input). Both charts have a time axis from 14:45 to 15:30.

Base de Datos en Firebase

The screenshot displays the Firebase console interface. The left sidebar contains navigation links for Project Overview, Authentication, Database (highlighted), Storage, Hosting, and Functions. Below these are sections for STABILITY (Crashlytics, Crash Reporting, Perf...) and ANALYTICS (Dashboard, Events, Audiences). At the bottom of the sidebar is a Spark section with the text 'Gratis 0 USD/mes' and a button labeled 'ACTUALIZAR'.

The main content area is titled 'Database' and shows the 'Realtime Database' selected. It includes tabs for DATOS, REGLAS, COPIAS DE SEGURIDAD, and USO. The URL bar shows 'https://console.firebase.google.com/u/0/project/blackmirrordb/database/blackmirrordb/data'. The database structure is displayed in a tree view under the path 'blackmirrordb > user'. The 'user' node contains several child nodes, with one expanded to show its details:

```
blackmirrordb
├── user
│   ├── -L1zHLTIL0vz2nDzGYkE
│   ├── -L1zHkgbwd_E_UVXrdcg
│   │   ├── Image: "url1"
│   │   ├── LastLocation
│   │   │   ├── Alt: 3456.7
│   │   │   ├── Lat: 233.3
│   │   │   └── Long: 445.4
│   │   ├── NRates: 0
│   │   ├── Name: "Hernan"
│   │   └── Rating: 3
│   └── -L1zHvt46uakJ0el1iG3
│       └── -L1z14kgQnRf8wz7chz7
```

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!