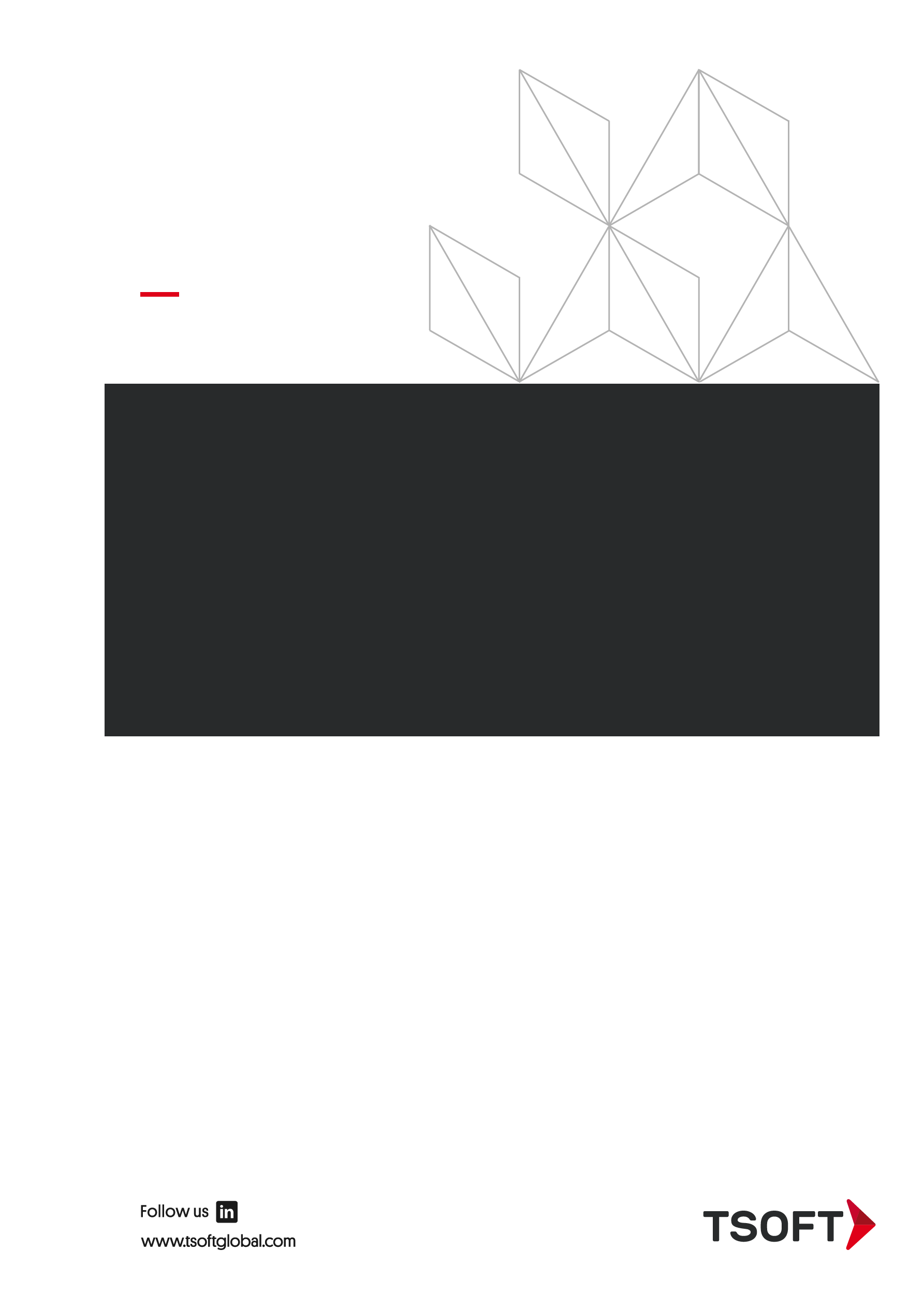
****

**Hubble**

***Administración***

**Diseño**

**Indice**

[1. Objetivos del documento 2](#__RefHeading___Toc196_141934952)

[2. Introducción 2](#__RefHeading___Toc801_3717703543)

[3. Seguridad 2](#__RefHeading___Toc840_3395260487)

[Autenticación 3](#__RefHeading___Toc180_1185719925)

[Autorización 3](#__RefHeading___Toc182_1185719925)

[4. Prototipos 3](#__RefHeading___Toc803_3717703543)

[5. Colecciones MongoDB 11](#__RefHeading___Toc805_3717703543)

# Objetivos del documento

Documentar el diseño de la aplicación de administración, el racional de las decisiones arquitectónicas y de diseño, los prototipos de las pantallas, y las colecciones MongoDB.

# Introducción

Las pantallas de administración se utilizarán para crear y configurar aplicaciones, configurar los *providers*, y administrar los usuarios de Hubble.

Se agregarán pantallas al *front-end*, a las que tendrán acceso los usuarios que tengan el rol Administrador.

Los fuentes de esta aplicación serán incorporados a los repositorios existentes de *front-end* y *back-end*, según corresponda.

El prototipo de las pantallas estará versionado en el repositorio de *back-end*, en el directorio “admin”.

# Seguridad

La seguridad tendrá un esquema simple de usuarios y roles.

Los usuarios deberán autenticarse en el sistema utilizando su dirección de correo electrónico, y su contraseña.

Los usuarios serán almacenados en la colección “userStorage” en MongoDB. La contraseña de cada usuario será almacenada en esa misma colección utilizando el algoritmo Argon2[[1]](#footnote-2). Este elige este algoritmo dado que está recomandado por OWASP[[2]](#footnote-3), existe una librería Java para utilizarlo.

Cada usuario podrá tener uno o más de los siguientes roles:

* Administrador: podrá acceder a todas las pantallas de administración de la aplicación.
* Usuario: podrá acceder a los dashboards del frontend.

El administrador podrá crear, modificar, y deshabilitar usuarios, así como blanquear las contraseñas.

## Autenticación

La autenticación se hará mediante un servicio REST, realizando un POST al recurso URL http://host:port/users/{email}/auth. El cuerpo del post deberá tener la contraseña.

La respuesta es de tipo JSON, con la siguiente estructura:

{

“token”: “”

“expirationDate”: “2018-06-30T00:00:00.00-3:00”

}

El token se almacena en el documento del usuario, dentro de la colección userStorage.

El token deberá enviarse en todas las invocaciones a servicios.

## Autorización

Para consultar la información del usuario, incluidos sus roles, realizar un GET a la URL http://host:port/users/{email}. La respuesta será de tipo JSON, con la siguiente estructura:

{

"id": "5b1abbb04f6e28305f4eaa7d",

"email": "admin@tsoftlatam.com",

"name": "Administrator",

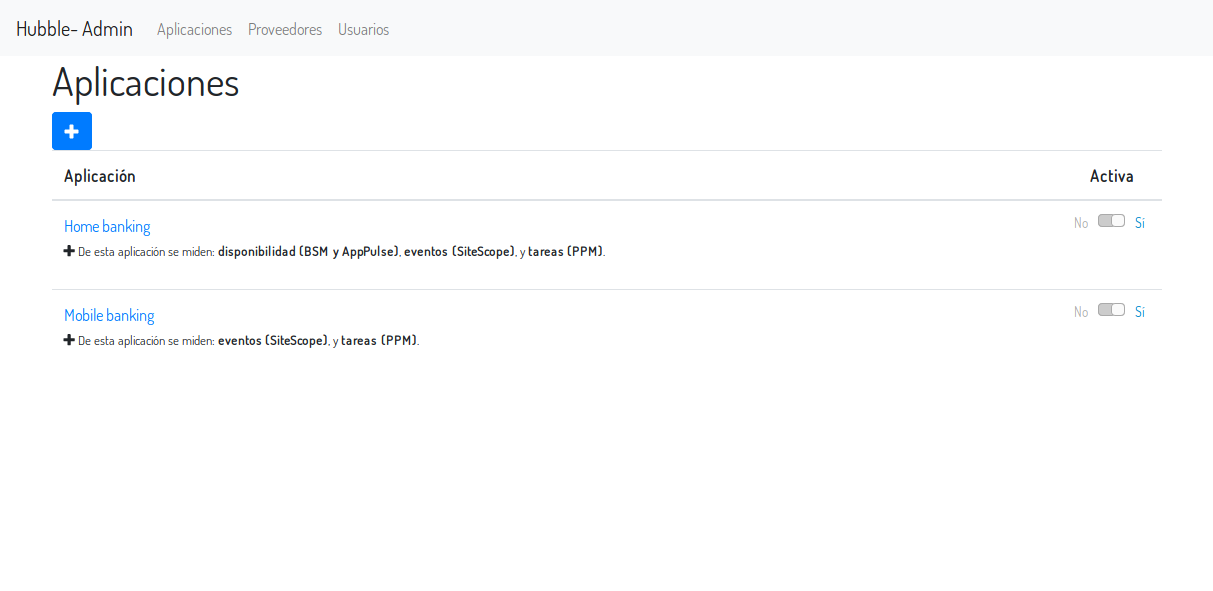
"roles": ["ADMINISTRATOR"],

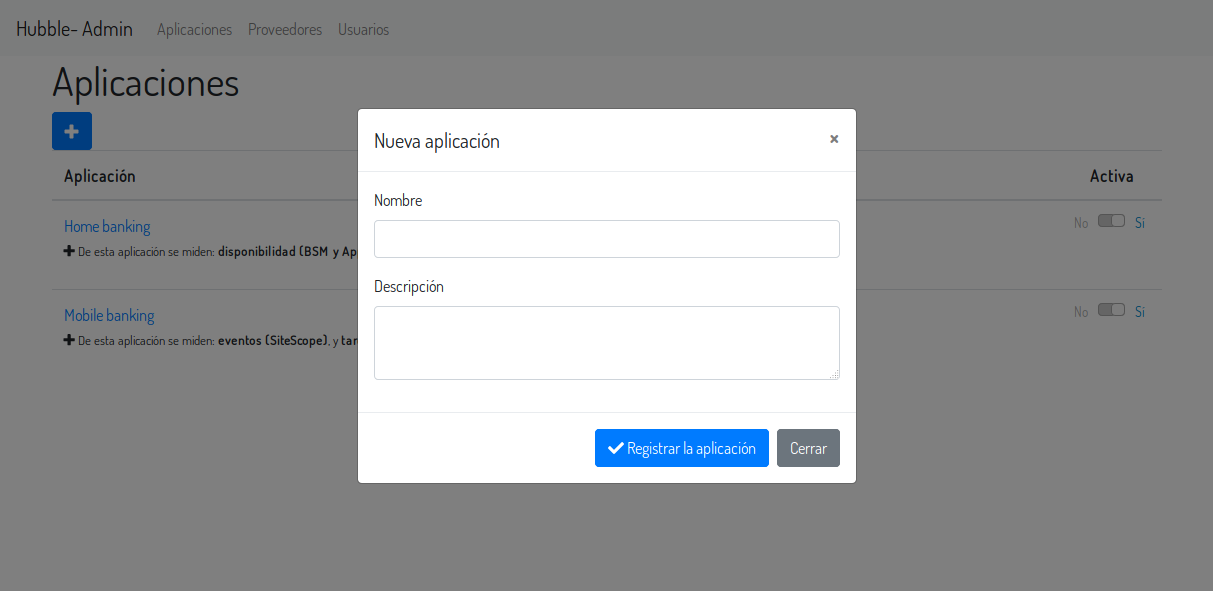
"applications": [“home\_banking”]

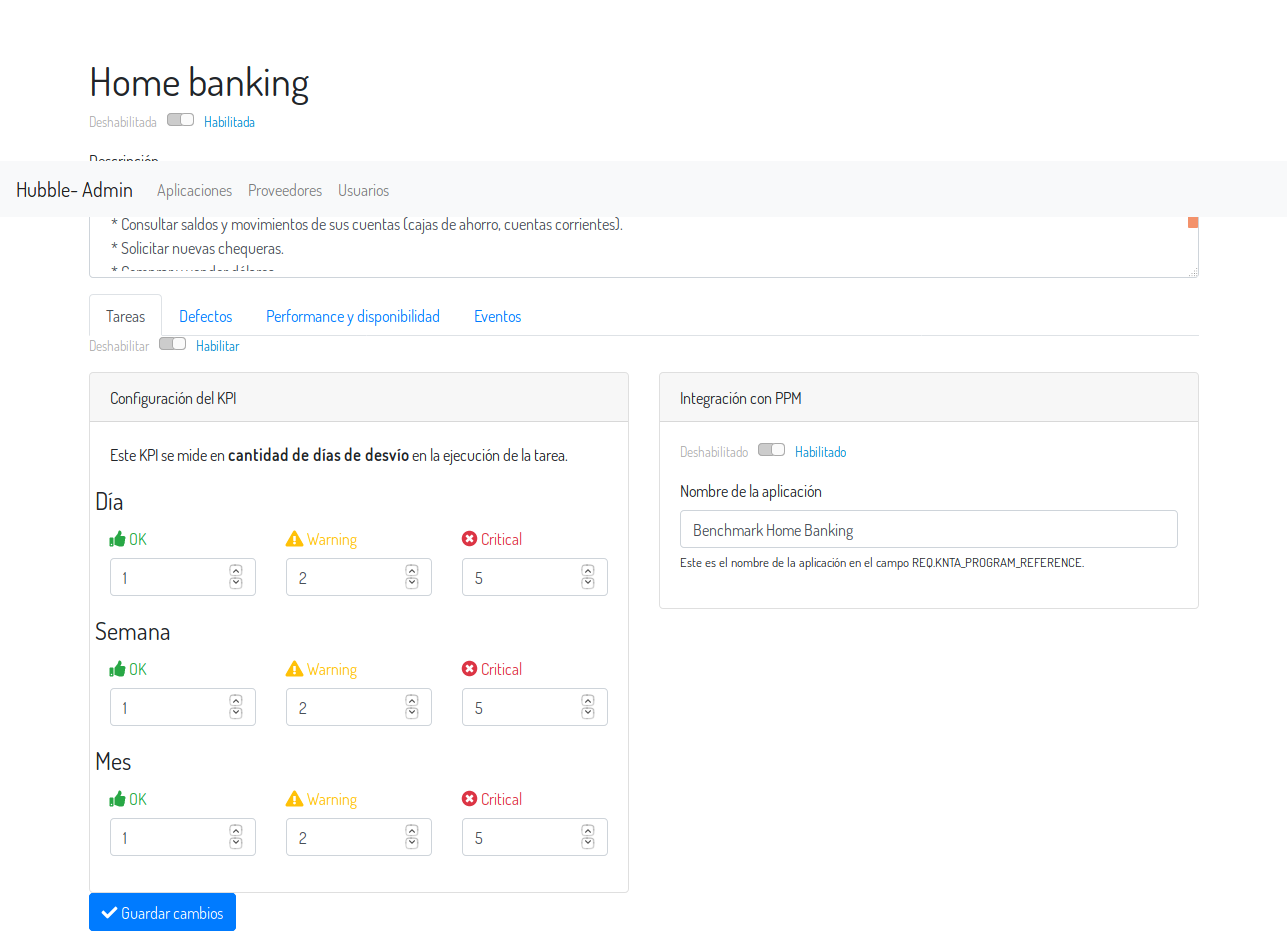
}

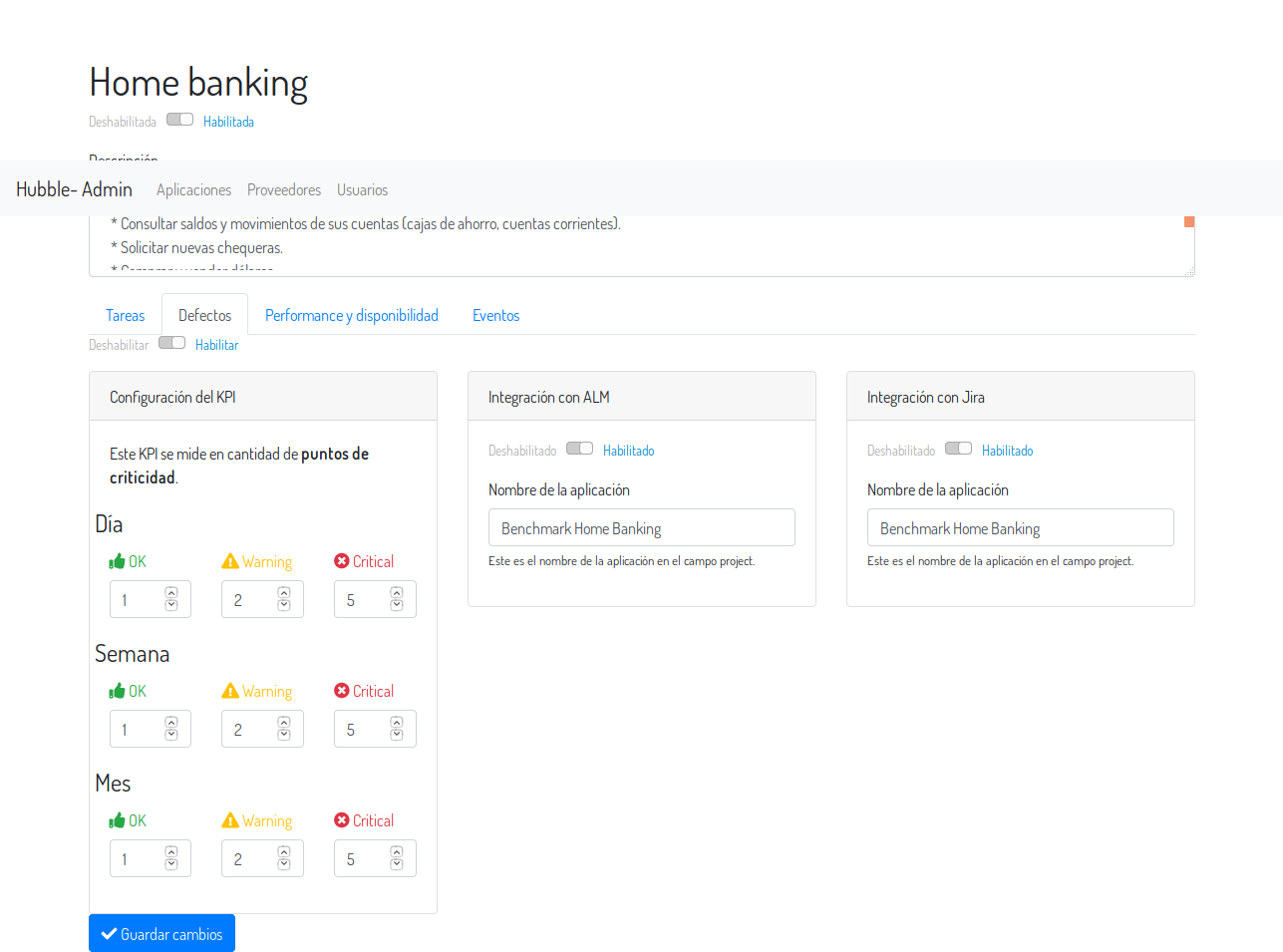
# Prototipos

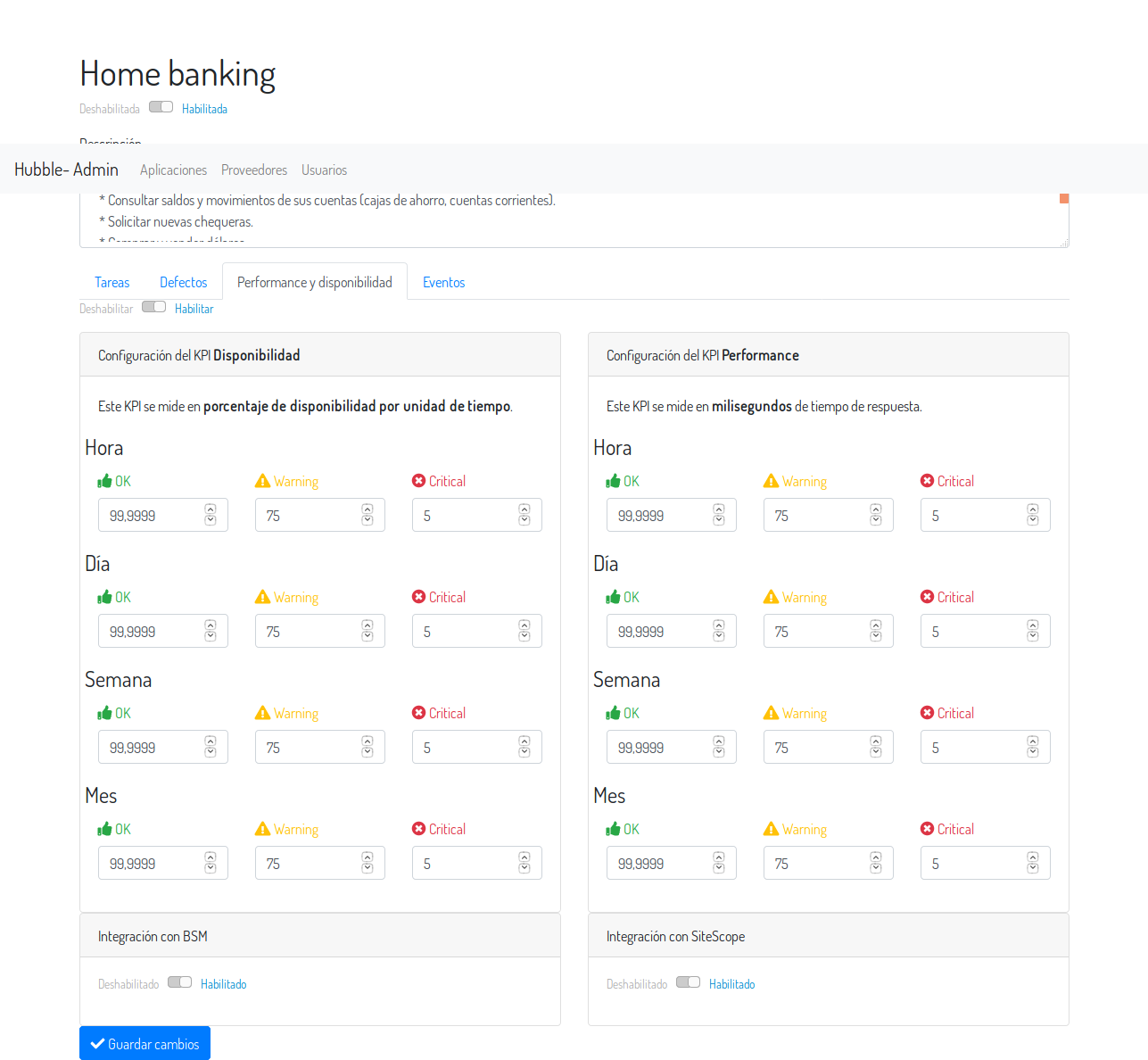
A continuación se incluyen los prototipos de las pantallas de administración.

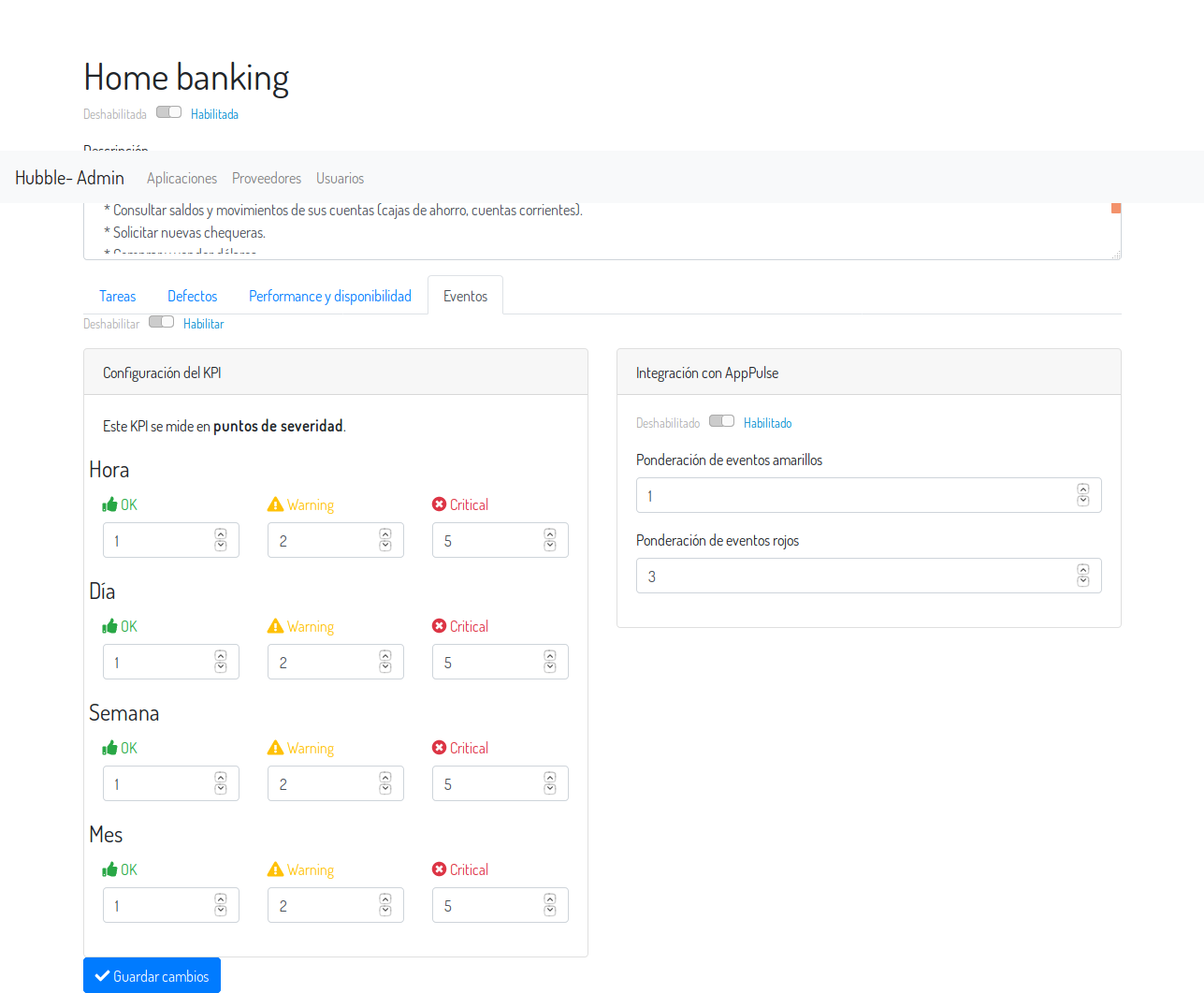


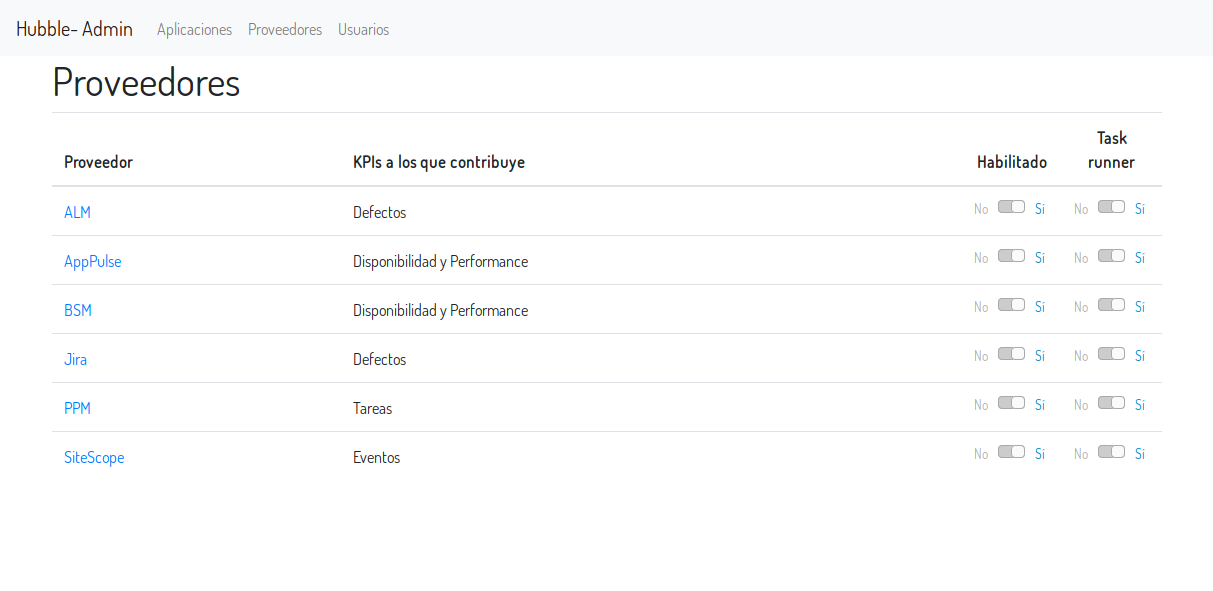


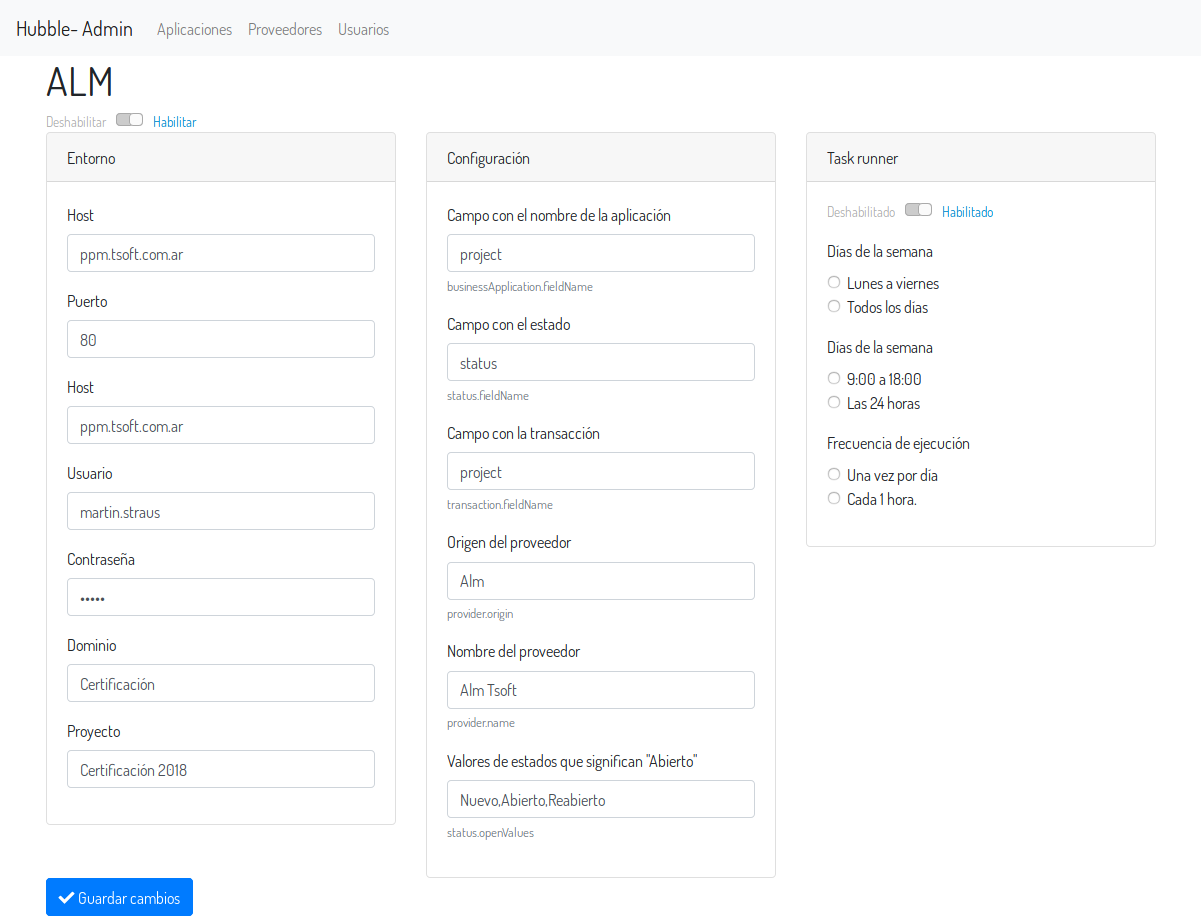


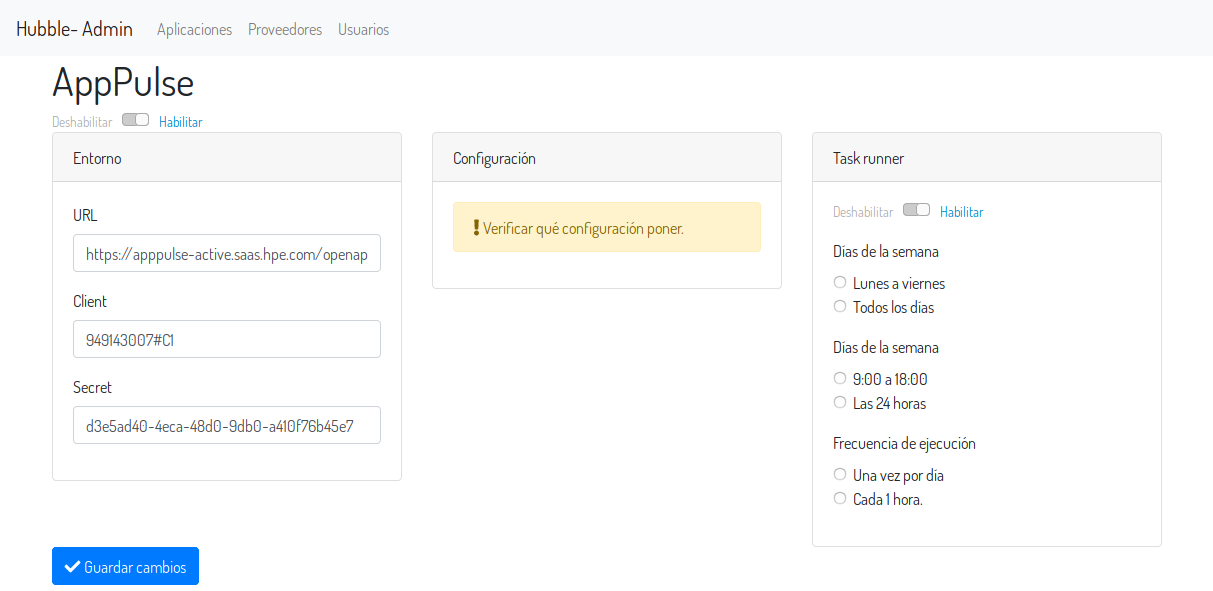


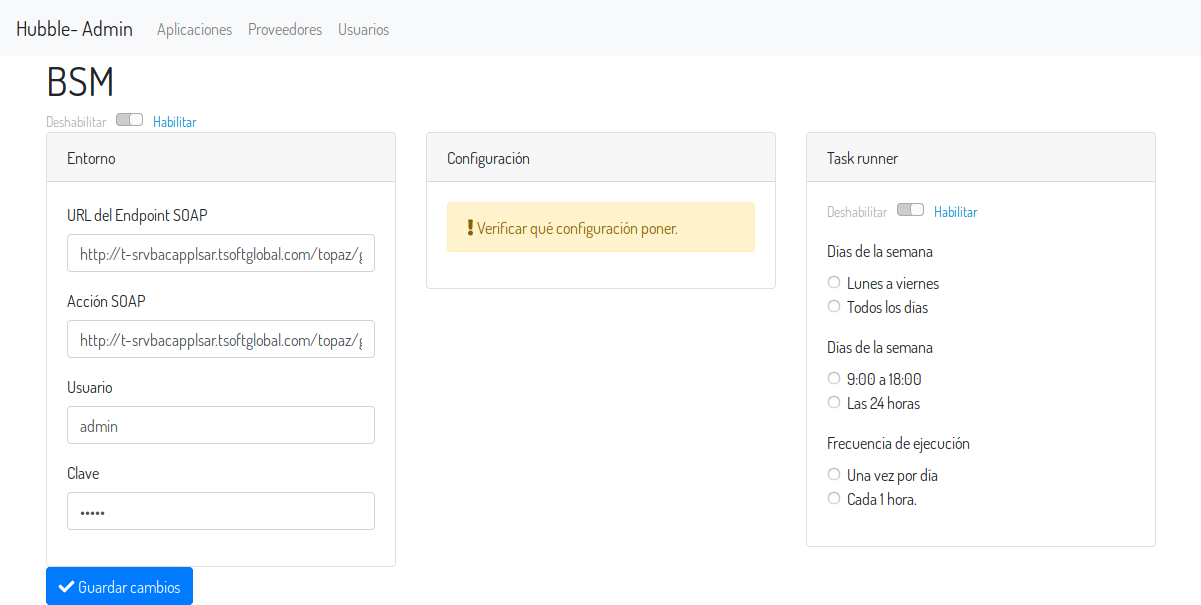


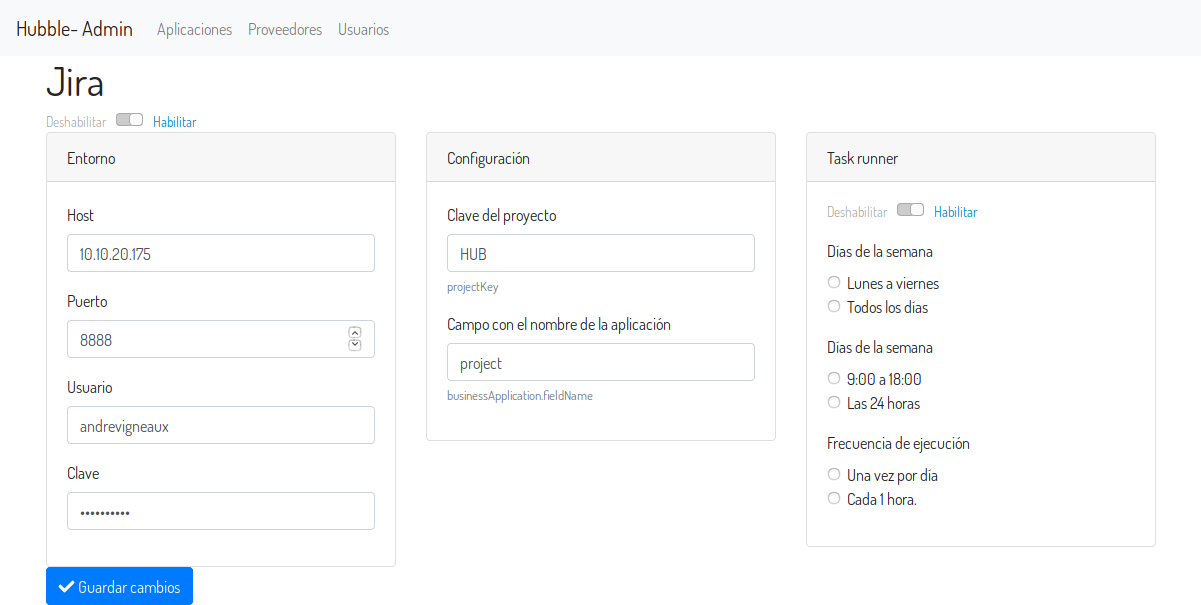


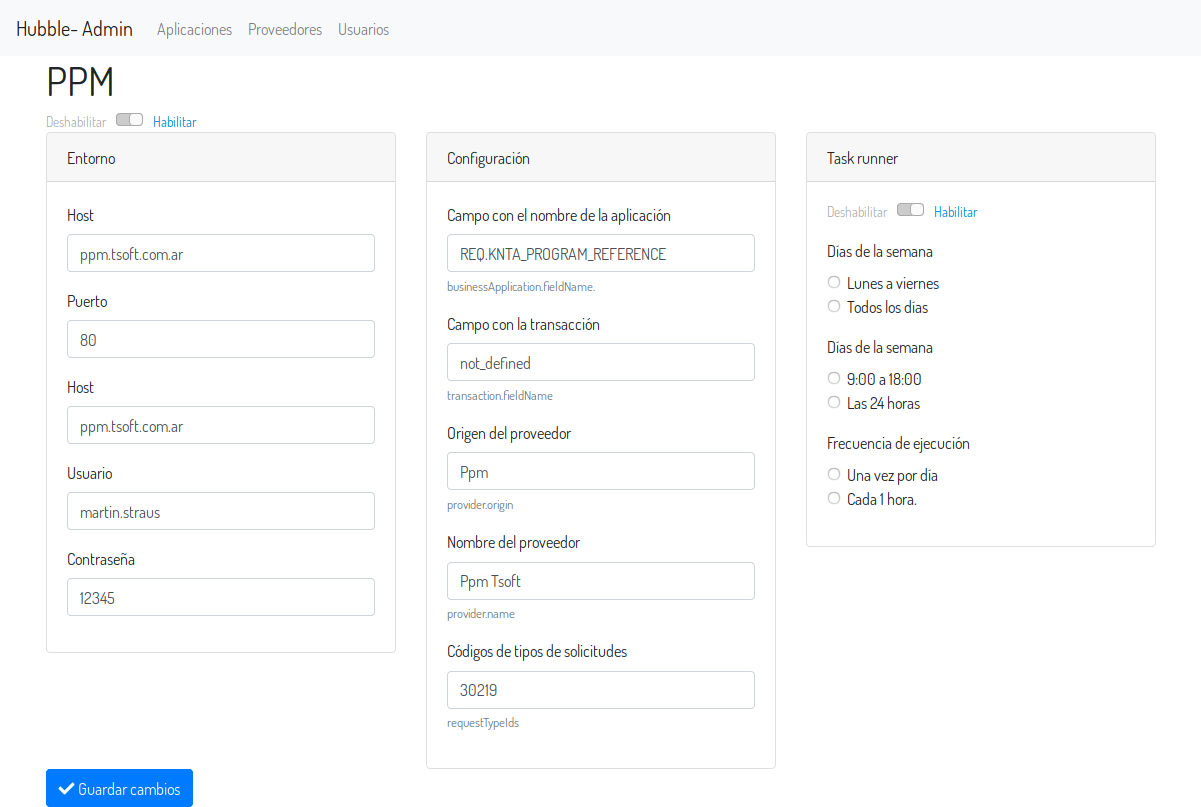


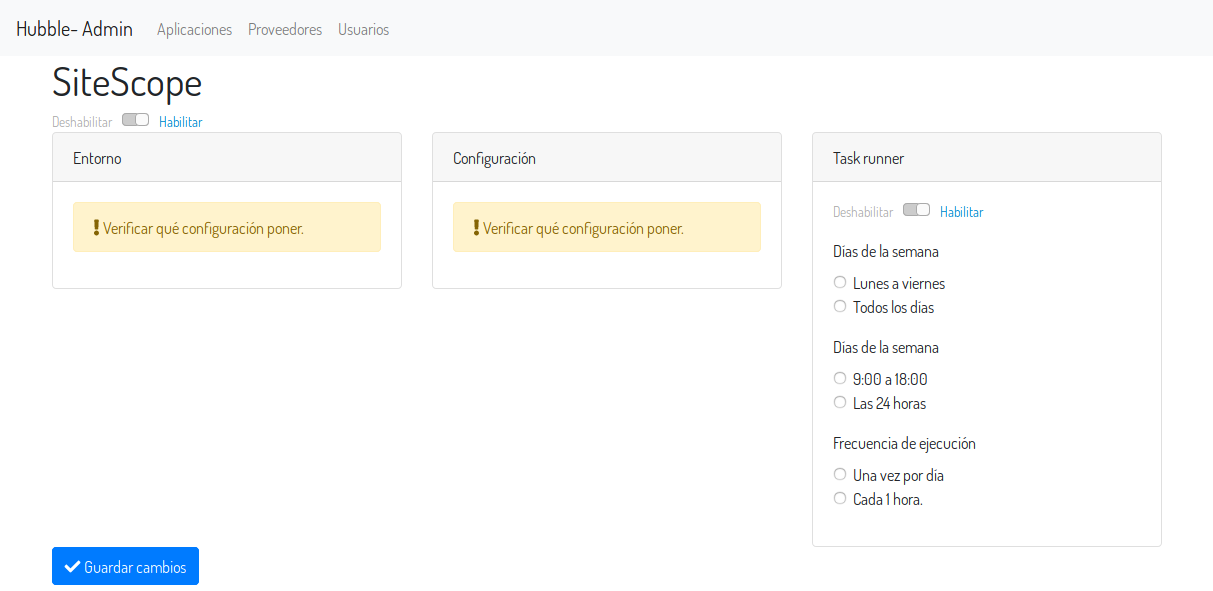


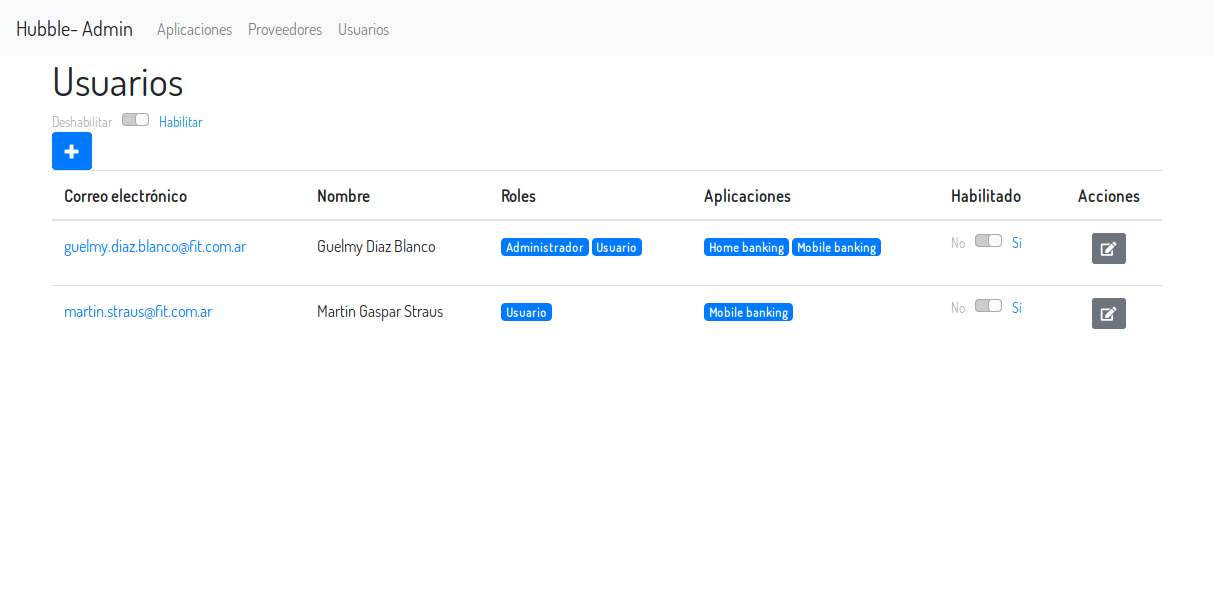


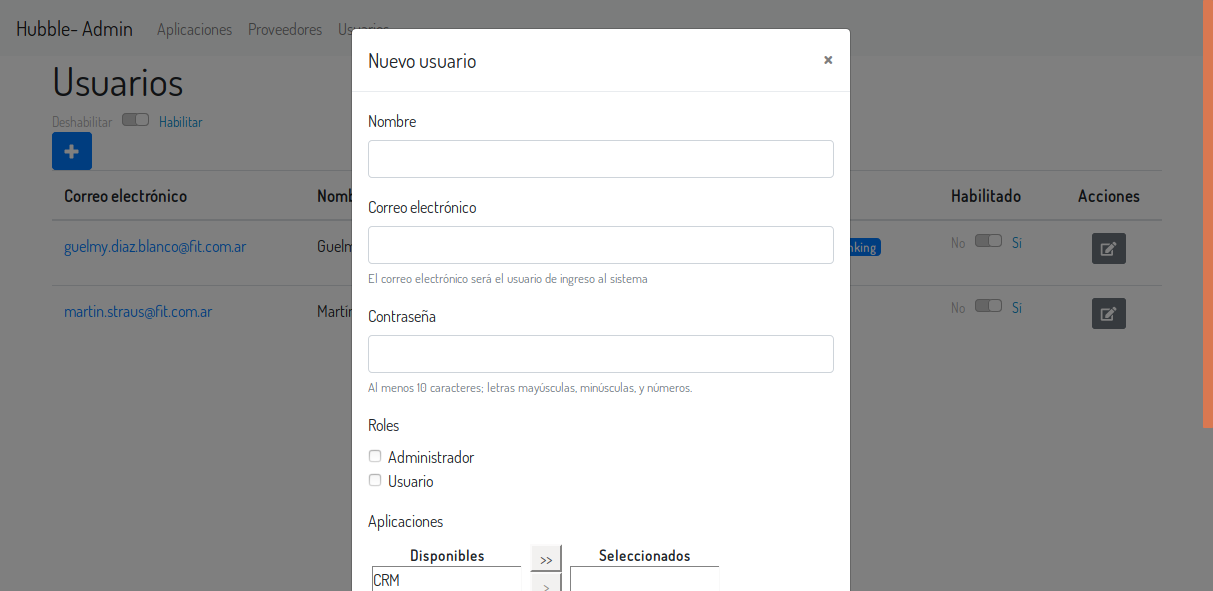


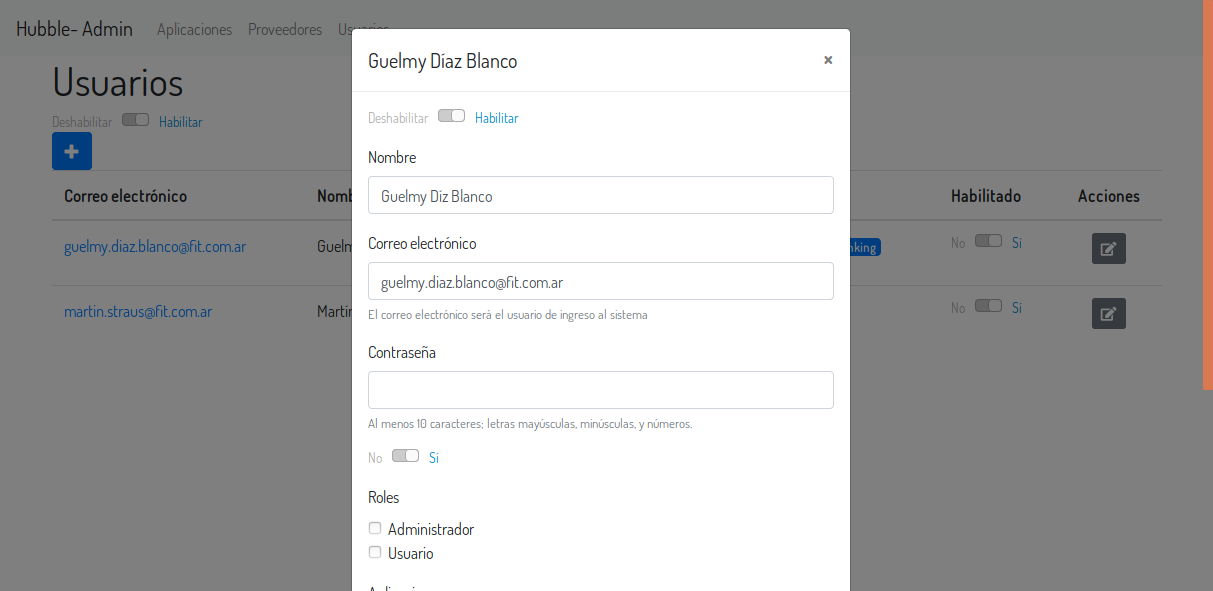






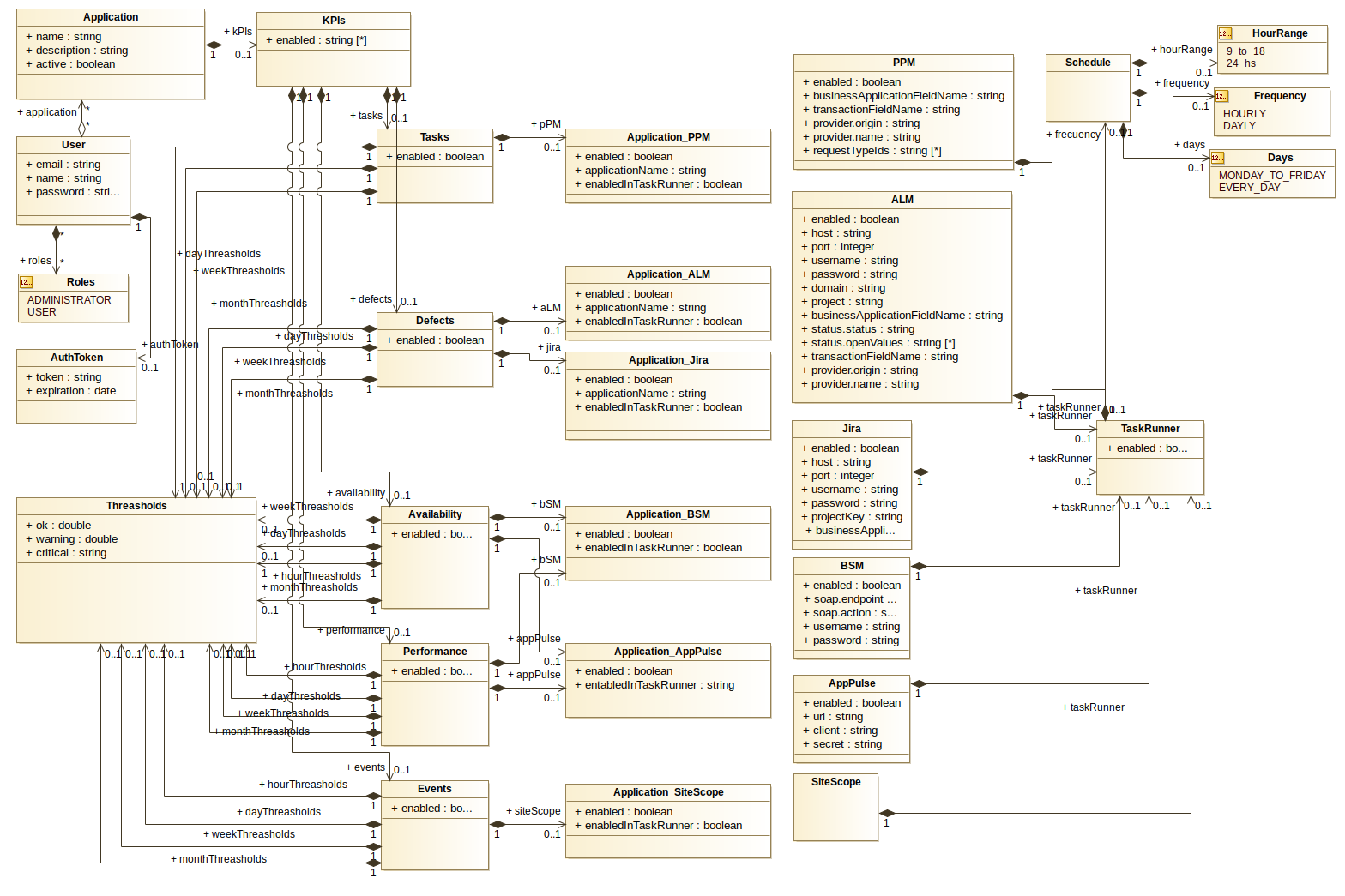






# Colecciones MongoDB

El siguiente diagrama representa todas las entidades agregadas o modificadas para las funciones de administración.



Las entidades serán almacenadas en las siguientes colecciones de MongoDB:

* applicationStorage
* providerStorage
* userStorage

1. <https://en.wikipedia.org/wiki/Argon2> [↑](#footnote-ref-2)
2. [https://www.owasp.org/index.php/Password\_Storage\_Cheat\_Sheet#Argon2\_usage\_proposal\_in\_Java](https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet" \l "Argon2_usage_proposal_in_Java) [↑](#footnote-ref-3)