**Proyecto**

“KeyBand”

**Grupo**

“11th”

"DOCUMENTO DE ESPECIFICACIÓN"

Hito: 0

Fecha entrega: 05-09-2016

Versión: 1.0

Componentes:

* Pedro Samuel Andreo Vilaplana
* Pablo Villar Canales
* Sergio Castro Sola
* Alberto García Garabal
* Fernando Gallud Mora
* Juan Sánchez Marín

**Contenido**

[Contenido](#_gjdgxs)

[1. Introducción](#_1fob9te)

[1.1. Propósito](#_3znysh7)

[2. Descripción general](#_2et92p0)

[2.1. Ámbito del sistema (Perspectiva del producto).](#_tyjcwt)

[2.2. Funcionalidades generales.](#_3dy6vkm)

[2.3. Características de los usuarios.](#_1t3h5sf)

[2.4. Interfaces externas.](#_4d34og8)

[2.5. Requisitos (suposiciones y dependencias).](#_2s8eyo1)

[2.6. Restricciones.](#_17dp8vu)

[2.7. Requisitos futuros.](#_3rdcrjn)

[3. Requerimientos específicos](#_26in1rg)

[3.1. Requerimientos funcionales.](#_lnxbz9)

[3.2. Requerimientos no funcionales.](#_35nkun2)

[4. Apéndices](#_1ksv4uv)

[4.1. Definiciones, acrónimos y abreviaturas](#_44sinio)

[4.2. Referencias](#_2jxsxqh)

[4.3.](#_2jxsxqh) Mockups

# Introducción

## Propósito

El objetivo de este documento es detallar la información, el propósito y las funcionalidades de nuestro sistema ‘KeyBand’. Este documento contiene todos los elementos de nuestro proyecto, desde restricciones hasta requerimientos tratándose de un manual para el correcto desarrollo de nuestro software ya que se detalla todo el sistema minuciosamente a través de aspectos como los requisitos, su validación, especificación y gestión, entre otros.

Este documento va dirigido a los profesores de las distintas asignaturas que engloba este ABP. Igualmente es de gran utilidad para nosotros ya que irá evolucionando iterativamente.

# Descripción general

## Ámbito del sistema (Perspectiva del producto).

‘KeyBand’ es un sistema que consiste en el control de acceso de usuarios en servicios hoteleros o similares. Igualmente proporciona soporte a la gestión de los diferentes servicios que se ofertan en los mismos. Diseñado para ser utilizado en web y app móvil, este sistema empleará unas pulseras dotadas con un chip NFC y unos lectores que se encargarán de interpretar los datos. Las aplicaciones para el móvil estarán disponibles tanto para Android como iOS.

Cabe destacar que ya existen en el mercado sistemas similares que utilizan la tecnología NFC para realizar diferentes tipos de acciones tales como ‘EasyGoBand’, que gestiona control de accesos y pagos, entre otros que se detallarán más adelante en el apéndice, pero ninguno que integre tantas funcionalidades como lo hace ‘KeyBand’. Es por ello que consideramos que nuestro producto es el más óptimo para abarcar esta parte del mercado.

‘KeyBand’ es un producto independiente ya que no forma parte de ningún otro sistema mayor. Sin embargo, el sistema podrá formar parte de un conjunto más extenso en el futuro.

## Funcionalidades generales.

‘KeyBand’ es un sistema enfocado a servicios hoteleros, cruceros u otros similares que facilita la capacidad de gestionar y monitorizar la actividad de sus clientes durante sus vacaciones, tales como gestionar el acceso a determinadas estancias a los clientes o las reservas que han hecho los mismos.

Por la parte de empleado, estos estarán divididos por diferentes tipos de usuarios con unos permisos concretos. Estos permisos abarcarán desde la creación de pulseras, el control de acceso a las diferentes estancias públicas o el registro de usuarios clientes del hotel, dependiendo del tipo de empleado en cada caso (recepcionista, camarero, masajista…).

A su vez, brindará a los clientes de estas empresas distintos servicios llevando encima una pulsera inteligente con NFC y una app para smartphone. Estas prestaciones incluyen desde acceder a habitaciones o estancias determinadas del establecimiento, hasta realizar pagos sin necesidad de llevar encima dinero en efectivo.

Además, todos los usuarios podrán visualizar el índice de ocupación de las diversas estancias públicas del establecimiento, definidas previamente por el administrador, ya que se llevará a cabo un control de acceso permanente. Para ello utilizaremos nuestro propio motor gráfico que cargará modelos 3D de las diferentes estancias.

## Características de los usuarios.

El sistema tendrá tres roles bien diferenciados. A continuación se especifican la formación y habilidades requeridas conforme a su función dentro del sistema.

**Administrador:**

* **Formación**: La formación del administrador deberá ser de nivel básico de informática pero medio-alto en el ámbito de la gestión.
  + **Habilidades**: El administrador tendrá el control absoluto sobre el sistema, la capacidad de poder gestionar todas las actividades/servicios del hotel además de poder conceder permisos a los empleados, asignar qué empleados van a realizar qué actividades, gestionar las estancias y acceder a las métricas y estadísticas.

**Empleado:**

* **Formación**: La formación del empleado que tendrá acceso a nuestro sistema bastará con tener un nivel básico para el manejo de la aplicación.
* **Habilidades**: Los empleados tendrán distintas funcionalidades dentro del sistema: capacidad de controlar accesos, dar de alta usuarios, gestionar las actividades que tienen asignadas, pedir soporte y establecer comunicaciones con los usuarios mediante notificaciones. Estas habilidades serán asignadas a los diferentes empleados en forma de permisos por parte del administrador, en función del rol que desempeñen dentro del hotel. El empleado, siempre que cuente con los permisos adecuados, podrá gestionar cualquier transacción o venta de productos mediante un TPV virtual integrado en la aplicación. Además, determinados empleados podrán comprobar, mediante un plano de calor 3D del hotel, la ocupación de determinadas estancias públicas.

**Cliente:**

* **Formación:** Bastará con que el cliente tenga una mínima experiencia en el manejo de aplicaciones con un nivel de usabilidad y accesibilidad estándar.
* **Habilidades:** los clientes podrán acceder a determinadas estancias públicas del hotel (gimnasio, piscina, spa…), además de a su estancia asociada (su habitación) mediante su pulsera con NFC. Tendrán a su disposición todas las actividades y productos ofertados por el hotel, además podrán pagarlos con solo acercar su pulsera a un lector siempre que tengan saldo suficiente. Los clientes del hotel podrán consultar métricas propias relacionadas con su actividad dentro del hotel (consumo, saldo, actividades a las que ha asistido…)

Mediante un servicio de notificaciones se garantizará la comunicación entre los distintos roles dentro del sistema. Estas notificaciones podrán ser diferentes avisos acerca de actividades, promociones, descuentos u otros.

## Interfaces externas.

En cuanto a interfaces externas asociadas a nuestro sistema, cabe destacar las siguientes:

* El sistema accederá a los servicios de la API de PayPal, a la hora de gestionar un pago o una transacción.

* La API de Facebook y Twitter permitirá la difusión de los diferentes servicios y productos que oferta el hotel, mediante tweets o publicaciones por parte del cliente. Esto será posible asociando la cuenta de usuario actual a la cuenta de Facebook o Twitter, una vez realizado el proceso de registro.

## Requisitos (suposiciones y dependencias).

Los dispositivos móviles de los empleados que se encarguen de gestionar cobros y el control de acceso a actividades/estancias han de disponer de la tecnología NFC y sistema operativo Android, ya que si no es imposible la interacción con las pulseras de los usuarios.

El hardware encargado del control de acceso (lectores) y del sistema de apertura de puertas debe ser compatible con NFC y ser capaz de hacer una llamada a una API.

Nuestro sistema estará alojado en la nube, de modo que se requerirá de conexión a Internet para acceder a los diferentes servicios, incluido el acceso a diversas API’s externas. Esto nos permitirá obtener beneficios, a modo de licencias mensuales, y cierto control sobre el uso del sistema hacia nuestros clientes.

## Restricciones.

La principal restricción que debemos tener en cuenta a la hora de desarrollar nuestro sistema son los ocho meses de los que disponemos para ello.

Por ello, se va a utilizar la metodología de desarrollo SCRUM para facilitar el flujo de trabajo e intentar agilizar los procesos.

También utilizaremos diversos frameworks y tecnologías que agilizarán el diseño de la parte front-end (como Angular2js), y el desarrollo multiplataforma (Ionic2).

Se diseñará una experiencia de usuario completa que favorezca y facilite la interacción del mismo con el sistema.

El sistema no guardará la tarjeta de crédito del usuario cliente, estos datos serán gestionados a través de la API de PayPal.

## Requisitos futuros.

- Sistema de control parental para supervisar las acciones de usuarios menores de edad, únicamente instaurado en usuarios con rol de cliente.

- Protocolos de seguridad más sofisticados ante un posible conflicto en el acceso a determinadas estancias, especialmente lugares abiertos, piscinas, parques…en los que el control de acceso con NFC pueda resultar más costoso.

- Función social en la aplicación: interacción entre los usuarios del hotel.

- Gestión de servicios de domótica y confort en las habitaciones, tanto desde dentro como desde fuera de ellas, tales como apagar y encender luces, aire acondicionado, etc.

# Requerimientos específicos

## Requerimientos funcionales.

Como bien se especificó en el punto 2.3, nuestro sistema será utilizado por diferentes usuarios a los cuales clasificaremos en tres roles diferentes (administrador, empleado y usuario), estos aparecerán detallados a continuación separados por qué pueden o no hacer en determinadas situaciones.

La parte del empleado es orientativa, cada empleado del hotel tendrá un rol diferente (camarero, recepcionista, monitor…) que les permitirá realizar unas funcionalidades u otras. El administrador de nuestro sistema será el encargado de otorgar a los empleados los diferentes permisos que habrán en cada hotel o crucero. En cada apartado se especifican todo aquello que pueden hacer los empleados en general.

**3.1.1 Requisito Funcional: Autenticación y gestión de cuentas.**

3.1.1.1 - Por parte del Administrador:

1 - Es capaz de gestionar todas las cuentas. Puede acceder al detalle de las cuentas de empleado y de usuario. También tiene permisos para buscar mediante filtros, crear, modificar o borrar estos tipos de cuenta.

2 - Da de alta y de baja las pulseras y las asigna a cada usuario.

3 - Puede aumentar el crédito de un usuario si este decide recargar en efectivo.

4 - Accede a la aplicación mediante un login y un password por defecto proporcionado por el desarrollador.

5 - Puede cambiar su contraseña e incluso utilizar la funcionalidad recuperación de contraseña. Con esto el sistema le enviará mediante un e-Mail un enlace para proceder al cambio de contraseña.

6- Es capaz de crear notificaciones y enviarselas a los cliente y a los empleados.

7- Asigna permisos a los diferentes roles de empleados.

3.1.1.2 - Por parte del Empleado:

1 - Accede a la aplicación mediante un login y un password por defecto proporcionado por el Administrador.

2 - Puede cambiar su contraseña e incluso utilizar la funcionalidad recuperación de contraseña. Con esto el sistema le enviará mediante un e-Mail un enlace para proceder al cambio de contraseña.

3 - Modifica sus datos personales.

4 - Puede crear, modificar, listar y eliminar usuarios clientes del hotel.

5 - Tiene la posibilidad de buscar y ver los detalles de usuarios como el régimen de su pensión.

6- Es capaz de crear notificaciones y enviarselas a los cliente.

3.1.1.3 - Por parte del Cliente:

1 - Accede a la aplicación mediante un login y un password por defecto proporcionado por un empleado del hotel.

2 - Puede cambiar su contraseña e incluso utilizar la funcionalidad recuperación de contraseña. Con esto el sistema le enviará mediante un e-Mail un enlace para proceder al cambio de contraseña.

3 - Modifica sus datos personales.

4 - Podrá asociar su cuenta en las principales redes sociales para posibles publicaciones (Facebook, Twitter)

**3.1.2 Requisito Funcional: Gestión y uso de estancias.**

3.1.2.1 - Por parte del Administrador:

1 - Crea, busca, modifica y elimina estancias dentro del sistema.

2 - Es capaz de controlar el acceso a una estancia con un Smartphone con NFC en caso de que el lector de una puerta determinada sufra una avería.

3.1.2.2 - Por parte del Empleado:

1 - Es capaz de controlar el acceso a una estancia con un Smartphone con NFC en caso de que el lector de una puerta determinada sufra una avería.

2 - Crea, busca, modifica y elimina estancias dentro del sistema.

3.1.2.3 - Por parte del Cliente:

1 - El usuario accede a las estancias mediante su pulsera con NFC KeyBand. En caso de que no esté autorizado a realizar esta acción, el sistema no permitirá el acceso.

**3.1.3 Requisito Funcional: Métricas y usos.**

3.1.3.1 - Por parte del Administrador:

1 - El administrador podrá observar estadísticas y datos de las horas de mayor aforo de las mismas estancias públicas especificadas en el punto 3.1.2.1.

2 - Puede ver estadísticas acerca de las actividades, productos o servicios consumidos por sus clientes durante la estancia en el hotel.

3 - Contará con estadísticas relacionadas con la ocupación del hotel, como en qué época del año se registran más clientes.

4 - Puede ver la ocupación en tiempo real de las estancias públicas del hotel a través de un mapa de calor 3D del mismo.

3.1.3.2 - Por parte del Empleado:

1 - Puede ver la ocupación en tiempo real de las estancias públicas del hotel a través de un mapa de calor 3D del mismo.

3.1.3.3 - Por parte del Cliente:

1 - El usuario podrá observar en todo momento aquel producto por el que ha pagado, viendo los tickets correspondientes por cada producto consumido o pudiendo ver el conjunto completo en una única factura.

**3.1.4 Requisito Funcional: Gestión de pulseras.**

3.1.4.1 Por parte del Administrador:

1 - El administrador podrá crear pulseras, modificar su estado y eliminarlas.

2 - Será capaz de listar todas las pulseras que han sido asignadas a un usuario independientemente de su estado.

3.1.4.2 Por parte del Empleado:

1 - El empleado también podrá crear pulseras, modificar su estado y eliminarlas.

2 - Será capaz de listar todas las pulseras que han sido asignadas a un usuario independientemente de su estado.

3 - Será capaz de cambiar el estado de una pulsera en un momento determinado, como por ejemplo en caso de pérdida de la misma.

3.1.4.3 Por parte del Cliente:

1 - Podrá acceder a las diferentes estancias y realizar pagos.

**3.1.5 Requisito Funcional: Gestión de productos y pagos.**

3.1.5.1-Por parte del Administrador:

1 - El administrador podrá dar crear productos, modificarlos (añadiendo fotografías descriptivas) y eliminarlos.

2 - Será capaz de listar todos los productos y las categorías de los mismos.

3 - Podrá controlar el pago de un producto o servicio con un dispositivo con NFC.

4 - Podrá asignarle a un empleado un producto de determinada categoría, como una excursión o un masaje.

5 - Será capaz de listar productos por usuario/empleado.

6 - En el caso de que un producto o una categoría de producto sea ofrecido en una estancia en concreto el administrador será encargado de asignarla.

3.1.5.2-Por parte del Empleado:

1 - Será capaz de listar todos los productos que han sido asignados a su categoría, es decir, un empleado podrá ver las excursiones a las que está asignado.

2 - Podrá listar los usuarios consumidores de un producto de una determinada categoría asignado a él, como por ejemplo, una excursión o un masaje.

3 - Podrá añadir crédito a un usuario si este desea el pago en metálico.

4 - Podrá controlar que se realice o se haya realizado el pago de un producto o servicio con un dispositivo NFC. Contará con para ello con un TPV en su aplicación.

3.1.5.3-Por parte del Cliente:

1 - Podrá ver todos los productos disponibles de determinada categoría tales como excursiones o clases aquagym.

2 - Podrá comprar productos a través de su app (excursiones, masajes, reservas de pistas de pádel…) descontando el importe de su saldo. En caso de ser insuficiente se le sugerirá una recarga por PayPal.

3 - Podrá añadir crédito a su cuenta a través de PayPal.

4 - Podrá solicitar el reembolso de su saldo.

5 - Podrá publicar en Twitter o Facebook una referencia positiva (tweet predeterminado por producto con posibilidad de modificación por parte del cliente) sobre un producto consumido favoreciendo la difusión de los servicios ofrecidos por el hotel o crucero.

**3.1.6 Requisito Funcional: Gestión de tickets.**

3.1.6.1-Por parte del Administrador:

1 - El administrador podrá acceder a todos los tickets de cada usuario.

2 - Será capaz de generar el ticket al realizar una venta de un producto, en consecuencia, creará líneas de ticket.

3.1.6.2-Por parte del Empleado:

1 - Será capaz de generar el ticket al realizar una venta de un producto, en consecuencia, creará líneas de ticket.

3.1.6.3-Por parte del Cliente:

1 - Podrá listar sus tickets, entrar en el detalle de cada línea de los mismos y ver los detalles del producto consumido.

2 - Podrá consultar y descargar en formato PDF o ePub cada uno de los tickets que haya realizado a través de diferentes compras a lo largo de su estancia. También tendrá la opción de reunirlos todos en una única factura.

**3.1.7 Requisito Funcional: Gestión del tablón de anuncios.**

3.1.7.1 -Por parte del Administrador:

1 - Podrá crear, editar y borrar anuncios con ayuda de un gestor de contenidos el cual facilitará dar formato, introducir contenido multimedia y añadir enlaces.

2 - Los anuncios podrá dirigirlos a todos los usuarios o solo a determinados segmentos de edad, tipos de clientes, etc.

3.1.7.2 -Por parte del Empleados

1 - Podrá crear, editar y borrar anuncios con ayuda de un gestor de contenidos el cual facilitará dar formato, introducir contenido multimedia y añadir enlaces.

2 - Los anuncios podrá dirigirlos a todos los usuarios o solo a determinados segmentos de edad, tipos de clientes, etc.

3.1.7.3 -Por parte del Cliente:

1 - Podrá visualizar en el tablón de su aplicación todos los anuncios de promociones, actividades, descuentos entre otros.

## Requerimientos no funcionales.

* 3.2.1. Requisitos de rendimiento

El 99% por ciento de las peticiones realizadas se realizarán en menos de un segundo. En caso contrario nuestro sistema mostrará un mensaje por pantalla, desde el cliente, avisando al usuario de que debe esperar unos segundos más mientras la operación solicitada se gestiona correctamente, lo cual generará interacción con el usuario y le suscitará una sensación de calma mientras se devuelve el recurso solicitado.

**3.2.1.1 Arquitectura Cliente/Servidor.**

El sistema seguirá una arquitectura cliente/servidor con estilo RESTful para el diseño e implementación de la API. Con ello se pretende que el sistema utilice un único punto de acceso para utilizar la API tanto por parte de la aplicación móvil como la de la aplicación web.

**3.2.1.2 Número de usuarios del sistema**

La aplicación debe garantizar el registro y uso de la totalidad de los clientes del hotel garantizando el acceso simultáneo a una gran cantidad de usuarios. En principio, se configurará un único servidor el cual tendrá capacidad para atender entre 100 y 150 usuarios. Más adelante se valorará la opción de incrementar el número de servidores y desarrollar un web-balancer que distribuirá las peticiones en función de los usuarios que tengan que acceder.

* 3.2.1. Seguridad
* La seguridad en el proceso de autentificación se va a basar en la implementación de un token de seguridad para garantizar la identidad del usuario.
* Cifrado del password del usuario.
* Cifrado del saldo disponible en la cuenta de un usuario.
* No se puede acceder a todos los datos del sistema a no ser que se haga desde la cuenta de Administrador.
* El acceso a los distintos servicios estará restringido por los diferentes roles que hay en nuestro sistema. Esto se controlará tanto en cliente como en servidor-
* El diseño de la API no reflejará la estructura de BBDD para evitar posibles inyecciones SQL.
* Las pulseras de uso serán solo de modo lectura.
* Todos los datos que introduzca el usuario e influyan en una petición a la base de datos pasarán por funciones que escapen los caracteres extraños evitando así las inyecciones SQL.
* 3.2.3 Fiabilidad

Pretendemos crear un sistema de auditoría para poder ser conscientes de en qué peticiones está fallando el sistema y poder reparar estos problemas.

* 3.2.4 Disponibilidad.

En caso de fallo de conexión a internet, se ofrecerán servicios mínimos instaurados en el servidor local, tales como la entrada y salida de estancias.

* 3.2.5 Mantenibilidad

El mantenimiento del software la realiza el desarrollador (nosotros), ya que tendrá acceso a estadísticas de uso y de fallos que le permitirá realizar mejoras.

De igual manera el usuario podrá pedir al desarrollador funcionalidades extras o informarle de modificaciones necesarias para el correcto funcionamiento del sistema.

El uso de métricas en el panel de administración, que se mostrarán tanto de forma semanal como mensualmente permitirá al desarrollador tener una aproximación de las carencias que pueda tener el sistema, qué funcionalidades explotar etc. Se llevará a cabo un mantenimiento mensual del sistema.

* 3.3.6 Portabilidad

Para que nuestro sistema sea portable y accesible desde cualquier plataforma, se construirá el sistema mediante tecnologías web, de modo que un amplio porcentaje de código estará comprendido entre: HTML5, CSS3 y Angular2js. Para la plataforma web y en el desarrollo del servidor también utilizaremos PHP para su creación.

Una vez construida la aplicación se distribuirá al resto de plataformas mediante Ionic2, un SDK de código abierto que trabaja encima de Angular2js y que permite la instalación de aplicaciones móviles gracias a una serie de herramientas de desarrollo y servicios.

# Apéndices

## Definiciones, acrónimos y abreviaturas

-NFC

***Near field communication, ‘****comunicación de campo’ cercano en español. Es la tecnología que usará el chip de la pulsera.*

-Smartband

***Pulsera inteligente*** *que se encarga de realizar las diversas funcionalidades a través de la transmisión de datos****.***

-PHP

***Pre Hypertext-processor,*** *lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.*

-PostgreSQL

***Sistema de gestión de base de datos relacional*** *orientado a objetos y libre que ofrece fiabilidad, integridad de datos y características.*

-Angular2.js

***F***[***ramework***](https://es.wikipedia.org/wiki/Framework) ***de*** [***JavaScript***](https://es.wikipedia.org/wiki/JavaScript)*de* [*código abierto*](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto)*, mantenido por* [*Google*](https://es.wikipedia.org/wiki/Google)*, que se utiliza para crear y mantener* [*aplicaciones web*](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)[*de una sola página*](https://es.wikipedia.org/wiki/Single-page_application)*.*

-Ionic2

***Nuevo framework*** *para crear aplicaciones web móviles basadas en HTML y poder hacerlas multiplataformas. Utiliza como base Cordova.*

-HTML

***HTML***, *sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web.*

-CSS

***Hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets)*** *es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).*

-Apache Server

***Servidor Web HTTP*** *de código abierto, modular, multi-plataforma y extensible.*

* SDK

***Software Development Kit*** *es generalmente un conjunto de herramientas de desarrollo software que permiten al programador crear aplicaciones para un sistema concreto.*

## Referencias

**TUI Smartband**

- Grupo TUI

* Control de accesos, pagos y luces.

<http://www.wareable.com/wearable-tech/new-hospitality-smartband-promises-a-better-holiday-experience-at-resorts-1962>

**Easy Go Band**

* Grupo PaynoPain Solutions S.L.
* Control de accesos, pagos y redes sociales.

<http://easygoband.com>

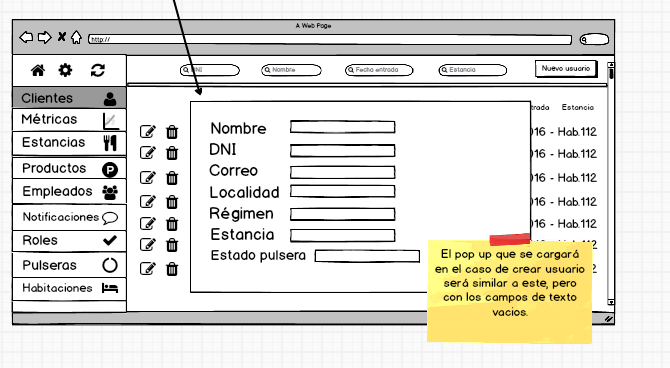
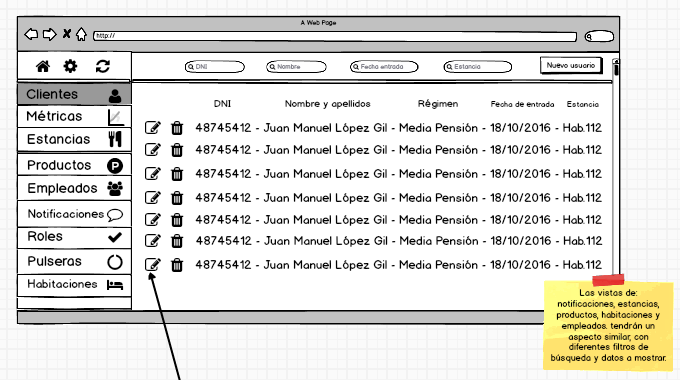
# **Smart VIB**

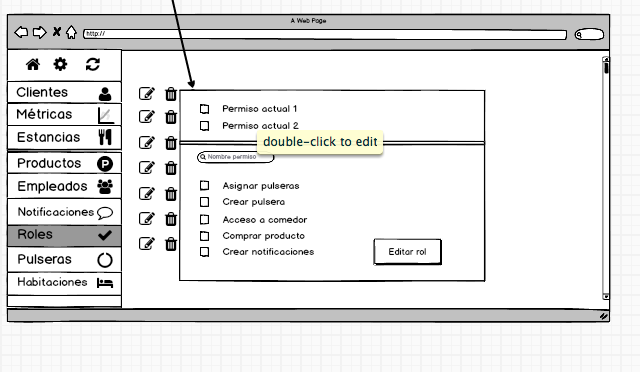
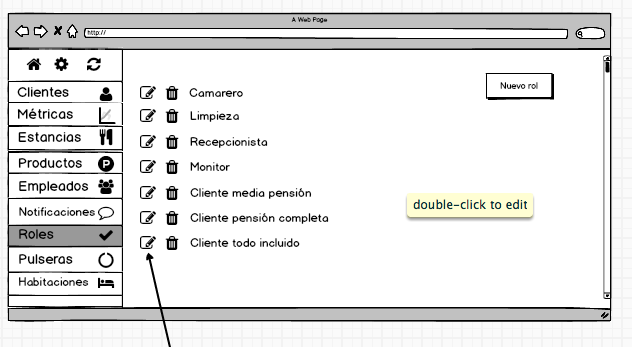
* Grupo Qing Solutions
* Control de accesos y pago

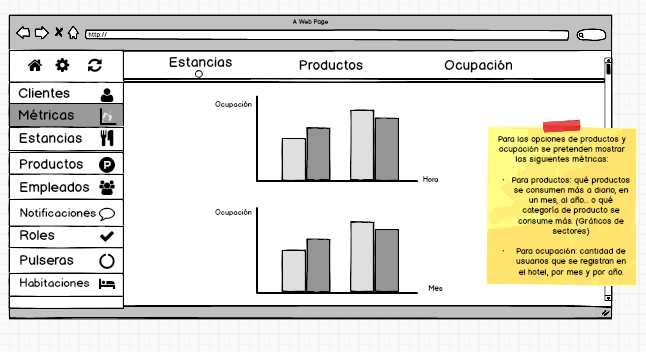
# <http://qing.solutions/tag/smart-vib/>

## Mockups

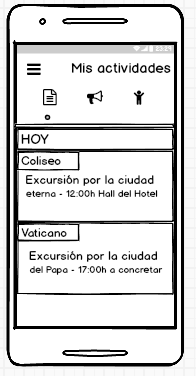
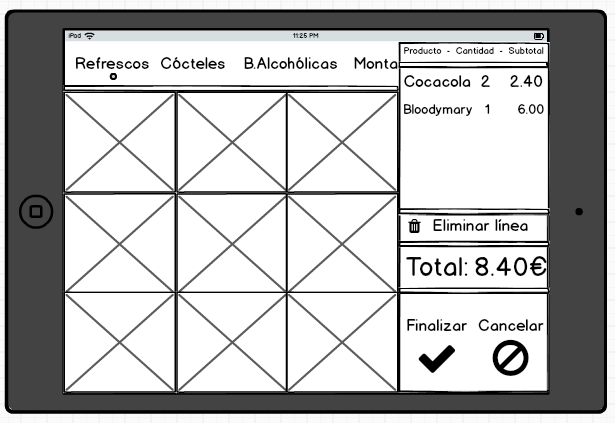
**MOCKUPS - Rol Administrador**



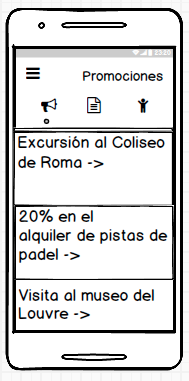
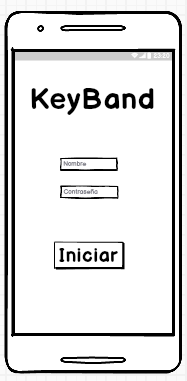


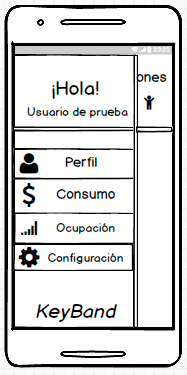
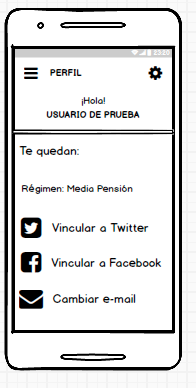


**MOCKUPS - Empleado**



**MOCKUPS - Rol Cliente**





4.4 Permisos asignables por rol.

* Crear usuarios
* Listar y detalle de usuario
* Editar usuario
* Borrar usuario
* Asignar Roles a usuario
* Asignar Pulsera a usuario
* Asignar Habitación a usuario
* Añadir crédito
* Dar de alta pulsera
* Modificar estado de pulsera
* Leer pulsera para ver el propietario
* Eliminar pulsera
* Crear productos
* Listar y detalle producto
* Editar producto
* Borrar producto
* Añadir publicación de Facebook predeterminada
* Añadir publicación de Twitter predeterminada
* Uso del TPV virtual y generación de tickets
* Crear estancias
* Listar y detalle estancias
* Editar estancias
* Borrar estancias
* Acceso a estancias como lector NFC
* Asignar permisos a rol
* Crear tickets
* Listar y detalle de tickets
* Editar tickets
* Borrar tickets
* Asignación de personal a actividades
* Crear notificación
* Listar y detalle notificación
* Modificar notificación
* Eliminar notificación