**Documento de requerimientos de software**

***Ticket app***

***Fecha: 30/07/2019***

**Tabla de contenido**

Historial de Versiones 3

Información del Proyecto 3

Aprobaciones 3

1. Propósito 4

2. Alcance del producto / Software 4

3. Referencias 4

4. Funcionalidades del producto 4

5. Clases y características de usuarios 5

6. Entorno operativo 5

7. Requerimientos funcionales 5

7.1. Administrar eventos 5

7.2. Administrar tickets 6

7.3. Administrar cupones 6

7.4. Administrar ventas 6

7.5. Administrar reportes 6

7.6. Mantenimiento perfil 7

7.7. Comprar tickets 7

7.8. Pagar ticket 7

7.9. Consultar tickets históricos 7

8. Reglas de negocio 8

9. Requerimientos de interfaces externas 9

10. Requerimientos no funcionales 9

11. Otros requerimientos 10

12. Glosario 10

# Historial de Versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 30/07/2019 | 1.0 | Miguel Ochoa Gallegos | Avantica | Creación de documento de requerimientos |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Información del Proyecto

|  |  |
| --- | --- |
| Empresa / Organización | Avantica techonologies |
| Proyecto | Ticket app |
| Fecha de preparación | 30/07/2019 |
| Cliente | Público en general |
| Patrocinador principal | Avantica techonologies |
| Gerente / Líder de Proyecto | Miguel Ochoa Gallegos |
| Gerente / Líder de Análisis de negocio y requerimientos | Miguel Ochoa Gallegos |

# Aprobaciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Propósito

Ticket app v1.0, tiene como propósito ser una plataforma online para crear cualquier tipo de evento así como administrarlo, desde la generación del evento: en donde se detalla categoría, precio entre otros, hasta la realización del pago por parte de los usuarios finales.

El proyecto nace a la necesidad de agilizar y automatizar la generación de eventos sin importar su tipología, brindando al usuario una experiencia novedosa de compra/venta de eventos.

El proyecto consta de una parte administrable, a la cual accederán usuarios específicos del tipo administrador y una parte operacional que será accedida por todos los usuarios interesados en los eventos mostrados.

# Alcance del producto / Software

El alcance del proyecto está en función a lo proporcionado por el proceso Everest.

# Referencias

**pmo:**

http://www.pmoinformatica.com

**Microservicios:**

Spring Microservices in Action.pdf

building-microservices.pdf

# Funcionalidades del producto

Las funcionalidades de la aplicación son:

1. Administrar eventos
2. Administrar tickets
3. Administrar cupones
4. Administrar ventas
5. Reportes
6. Mantenimiento perfiles
7. Realizar compra/pago de tickets
8. Consultar compras históricas

# Clases y características de usuarios

Para la aplicación de Ticket app, se definieron las siguientes clases de usuario por funcionalidad:

* **Administrador**: encargado de administrar los eventos, tickets, cupones, ventas, perfiles y reportes
* **Operativo**: encargado de realizar las tareas de compra, consultas históricas y administración de su propio perfil

# Entorno operativo

La aplicación será construida usando tecnologías open source:

Back-end: se usará java para creación de servicios y microservicios.

Front-end: para la creación de la ui para los diferentes tipos de usuarios. (por definir entre vue o angular)

Git: para el versionamiento de fuentes

Además de ser una aplicación web, será desplegada usando tecnología cloud (por definir heroku o aws)

# Requerimientos funcionales

Para la realización del proyecto se listan los siguientes requerimientos funcionales:

# Administrar eventos

* La aplicación mostrará la opción de administrar eventos para el perfil: administrador.
* La aplicación mostrará las opciones: crear evento, modificar evento, cerrar evento.
* La aplicación mostrar los diferentes tipos de eventos, así como su mantenimiento respectivo.
* La opción crear evento: solicitará la información fecha del evento, costo de entrada, asignación de cupones de descuento si existiese, dirección del evento, así como otros datos importantes y que definan al evento.

# Administrar tickets

* La aplicación mostrará la opción de administrar tickets para el perfil: administrador.
* La aplicación mostrará las opciones: crear tickets, modificar tickets, cerrar tickets.
* La opción crear tickets: solicitará el número de tickets a generar, en función al aforo que tenga el evento creado, así como la relación con la entidad evento.

# Administrar cupones

* La aplicación mostrará la opción de administrar cupones para el perfil: administrador.
* La aplicación mostrará las opciones: crear cupones, modificar cupones, cerrar cupones.
* La opción crear cupones: solicitará el porcentaje de descuento del cupón, fecha de validez del cupón, número de cupones, así como la relación con la entidad evento.

# Administrar ventas

* La aplicación mostrará la opción de administrar ventas para el perfil: administrador.
* La aplicación mostrará la consulta de los pagos realizados por evento
* La aplicación mostrará el detalle de la compra efectuada por ticket y evento, además de los descuentos de tener un cupón.
* La aplicación notificará vía correo, 1 día antes del evento, el detalle de la venta de los tickets

# Administrar reportes

* La aplicación mostrará la opción de administrar reportes para el perfil: administrador.
* La aplicación mostrará las opciones: consulta de ventas por evento
* La aplicación mostrará las opciones: consulta de mes de venta
* La aplicación mostrará un ranking de eventos generados en función a la venta que tuvieron.
* La aplicación mostrará la opción de descargar el resultado.

# Mantenimiento perfil

* La aplicación mostrará la opción de administrar perfiles para el perfil: operativo.
* La aplicación mostrará la opción de crear usuario, en donde solicitará datos de contacto, datos de pago: tarjeta de crédito o cuenta paypal
* La aplicación mostrará tipos de eventos del gusto del usuario, para notificarlo via correo, cuando este cerca a fecha de realización

# Comprar tickets

* La aplicación mostrará la opción de comprar tickets para el perfil: operativo.
* La aplicación mostrará los eventos ordenados por fecha de realización
* La aplicación los cupones de descuentos de los diferentes eventos creados

# Pagar ticket

* La aplicación mostrará la opción de pagar tickets para el perfil: operativo.
* La aplicación mostrará pasarela de pagos para el evento elegido
* La aplicación mostrará constancia de compra del ticket
* La aplicación enviará un correo electrónico con la constancia de la compra realizada.
* La aplicación enviará un correo recordatorio 1 día antes del evento al usuario.

# Consultar tickets históricos

* La aplicación mostrará la opción de consulta histórica de tickets para el perfil: operativo.
* La aplicación mostrará los diferentes filtros para realizar la consulta.
* La aplicación mostrará el resultado de la búsqueda, ordenados por fecha de creación.
* La aplicación mostrará paginación en caso el nro de tickets sea mayor al configurado.
* La aplicación mostrará la opción de realizar la descarga de la consulta histórica.
* La aplicación mostrará la opción de realizar una nueva búsqueda.

# Reglas de negocio

Las reglas de negocio definidas en el inicio del proyecto son:

* No existe mantenimiento para usuario tipo administrador.
* No existe ningún evento que no esté tipificado.
* No existe ningún ticket que no esté asociado a un evento.
* No existe ningún cupón que no esté asociado a un evento.
* El usuario que compre un ticket, no podrá realizar una cancelación.
* Los cupones de descuento no son acumulativos ( son de uso único por evento)

# Requerimientos de interfaces externas

Ninguno.

# Requerimientos no funcionales

Los requerimientos no funcionales fueron organizados como se muestra a continuación:

**Eficiencia**

La aplicación debe ser capaz de procesar N transacciones por segundo. Esto se medirá por medio de la herramienta Jmeter.

Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.

Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de N segundos.

**Seguridad lógica y de datos**

La aplicación debe desarrollarse aplicando patrones de seguridad que incrementen la seguridad de datos.

Las comunicaciones entre servicios dentro de la aplicación deben tener un filtro de encriptación para evitar ataques.

**Usabilidad**

El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas. El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.

La aplicación web debe poseer un diseño “Responsive”.

El sistema debe poseer interfaces gráficas bien formadas.

# Otros requerimientos

Ninguno.

# Glosario

**Microservicios**: es la separación de un clásico servicio monolítico por muchos servicios que tiene un tarea bien definida.

**Vue**: librería js que simplifica el uso de html, css y js en el mismo documento

**Angular**: framework front end, creado por google para el desarrollo de ui modernas

**Heroku**: es una plataforma de la computación en la nube.

**Aws**: Amazon web services, es una infraestructura/plataforma de la computación en la nube.

**Jmeter**: herramienta para testeo de concurrencias para servicios rest

**Responsive**: que se adaptable en los diferentes dispositivos: escritorio, Tablet, celular, etc.