

**Control de Versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Autor** | **Versión** | **Descripción** | **Fecha** |
| David Cortes | 1.0.0 | Versión final de documento de arquitectura de Microservicios con Java y Spring Boot | 15/03/2024 |

**Tabla de Contenidos**

1. Decisiones de Arquitectura 4
2. Atributos de Calidad 5
3. Visión de Arquitectura 6
4. Escenarios 8
5. Vista de casos de uso 12

​​

|  |  |  |
| --- | --- | --- |
| **Número** | **Título** | **Descripción** |
| D-001 | Estilo de Arquitectura | La solución tomará como estilo el patrón SOA. |
| D-002 | Infraestructura Base | A nivel de infraestructura, se decidió apoyarse en la plataforma AWS de Amazon. |
| D-003 | Patrón complementario - Microservicios | El uso de microservicios como patrón complementario nos permite la distribución de tareas en pequeños módulos que sean independientes entre sí. |
| D-004 | Patrón complementario: AMQP | Se propone un patrón AMQP para la alineación de las bases de datos de cada servicio. |
| D-005 | Patrón complementario: JWT Authentication | Se propone una autenticación usando JWT, el cual autenticara cada endpoint. |
| D-006 | Bases de Datos | Se utilizará un database as a service, cada servicio tendrá su propia base de datos independiente. |

1. Decisiones de Arquitectura

1. Atributos de Calidad

**Adecuación Funcional**

* **Corrección Funcional:** Debido a la naturaleza de las operaciones que se llevarán a cabo dentro de la plataforma se hace necesario que el sistema cumpla a cabalidad las funciones para las que fue desarrollado.

**Eficiencia de desempeño**

* **Comportamiento temporal:** La solución tecnológica debe presentar tiempos de respuesta bajos en relación a los portales de la competencia, con la intención de mantener a los usuarios dentro del flujo de compra.

**Fiabilidad**

* **Disponibilidad:** Con el fin de mantener un alto índice de disponibilidad del sistema nos centraremos en *Redundant Implementation y Service Grid*, apoyado en funcionalidades brindadas por el proveedor Cloud escogido.

**Seguridad**

* **Confidencialidad:** Debido a la naturaleza de las transacciones que se procesaran, las que involucran operaciones con cuentas y tarjetas, se asegurara las mismas con autenticación basada en JWT.
* **Autenticidad:** Se debe garantizar que el 100% de las comunicaciones fuera de las fronteras del sistema sean mediante el protocolo cifrado *HTTPS*. Dentro del entorno del sistema (zona desmilitarizada) se usarán comunicaciones seguras en base a Trusted Subsystem.

**Usabilidad**

* **Operabilidad**: La solución debe permitir al usuario operarla y controlarla con facilidad de forma que sea ágil la creación de un plan turístico.

**Compatibilidad**

* **Interoperabilidad:** La solución debe intercambiar información con proveedores como empresas de transporte y hoteles y permitir agregar más proveedores en cualquier momento por lo que requiere un alto grado de interoperabilidad.

**Mantenibilidad:**

* **Capacidad de ser modificado**: La solución debe permitir que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño con el fin de permitir que la integración con las plataformas de los proveedores sea flexible

1. Visión de Arquitectura

Brindar a la entidad financiera una solución tecnológica orientada a la mejora continua de sus procesos, la satisfacción de sus clientes y el crecimiento orgánico del negocio, basándose en el empoderamiento del área de IT, la expansión de los proveedores y un flujo de seguimiento al usuario antes, durante y después de su compra.

2.1 Árbol de Utilidad (SMART - Specific - Measurable - Achievable – Real Time Constraint)

A = Alto, M = Medio, B = Bajo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo de Calidad | Criterio | Id Escenario | Descripción Escenario | Imp. | Dif. |
| Comportamiento temporal | Tiempos de ejecución para consulta de disponibilidad. |  | Se debe garantizar que el sistema tenga bajos tiempos de respuesta al momento de consultar la disponibilidad de los diferentes productos ofrecidos. | M | A |
| Comportamiento temporal | Tiempos de ejecución para la creación de reservas |  | Se debe garantizar que el sistema tenga bajos tiempos de respuesta al momento de consultar la disponibilidad de los diferentes productos ofrecidos. | A | A |
| Interoperabilidad | Comunicación con diferentes proveedores |  | Se debe garantizar la flexibilidad para le intercambio de información con otros sistemas y la agregación de nuevos sistemas sin necesidad de reiniciar el sistema. La integración de nuevos proveedores se debe realizar en un máximo de una semana. | A | M |
| Seguridad | Autenticidad |  | Las solicitudes realizadas deben tener un header de autenticación. | M | B |
| Capacidad de ser modificado | Agregar nuevas funcionalidades o integrar nuevos proveedores |  | Garantizar la modificación del sistema de forma efectiva y eficiente sin introducir defectos o degradar el desempeño | A | A |
| Capacidad de ser modificado | Nuevas funcionalidades |  | Desplegar nueva funcionalidad sin afectar la funcionalidad actual y en un tiempo máximo de 4 semanas hombre | A | M |

1. Escenarios

|  |  |
| --- | --- |
| **Escenario** | Respuesta del Sistema |
| **Justificación** | Al revisar los tiempos de la competencia se nota que es necesario implementar un sistema con tiempos de respuesta lo más cercano posible a los tiempos de la industria. |
| **Atributos de Calidad** | Corrección funcional, Comportamiento temporal, Disponibilidad |
| **Estímulos** | Interacción del usuario con la página. |
| **Respuesta** | Bajo condiciones de baja carga y carga normal la respuesta del sitio no debe sobrepasar los 4 segundos por petición. En caso de alta carga se permite la degradación del sistema respecto al tiempo de carga pero no al tiempo por petición |
| **Decisiones de Arquitectura** | Utilizar un modelo de nube elástica. A nivel de AWS, un farm autoescalable basado en servidores t2.medium sirve para tráfico bajo a intermedio. En el caso de punto pico, lo ideal es escalar a instancias t2.large Ref: https://aws.amazon.com/ec2/instance-types/ |
| **Medición** | Tiempo de respuestas en segundos. |

|  |  |
| --- | --- |
| **Escenario** | Usabilidad del Sistema |
| **Justificación** | Al consistir en sistema de auto-servicio, es necesario medir la satisfacción del cliente. |
| **Atributos de Calidad** | Corrección funcional, Operabilidad |
| **Estímulos** | Interacción del usuario con la página. |
| **Respuesta** | La interfaz de usuario debe cumplir con ser intuitiva y responsiva |
| **Decisiones de Arquitectura** | El uso de Angular y Bootstrap nos permite mantener flujos rápidos de respuesta de cara al usuario y portabilidad entre dispositivos. |
| **Medición** | Promedio de tiempo en la página.  Porcentaje de Rebote.  Embudo de conversión. |

|  |  |
| --- | --- |
| **Escenario** | Respuesta a ataques |
| **Justificación** | Al estar en un entorno en el que se manejan transacciones financieras es necesario tener un robusto esquema de seguridad y respuestas claras frente a ataques de terceros. Además, es necesario que los ataques no generen sobrefacturación por aspectos como el autoescalamiento de la infraestructura. |
| **Atributos de Calidad** | Rama de seguridad, Disponibilidad. |
| **Estímulos** | Ataques programados de terceros. |
| **Respuesta** | Las interacciones con los usuarios deben seguir un estricto protocolo de seguridad. Empezando con autenticación segura vía tokens y respaldada por una conexión HTTPS. A nivel de ataques, es necesario que estos sean detenidos tan pronto como sean detectados y con el menor impacto posible en la arquitectura. |
| **Decisiones de Arquitectura** | Utilizar un patrón tipo API Gateway de cara al exterior blinda en gran medida los accesos no autorizados a la arquitectura. A nivel de solución cloud, el soporte de los productos AWS Shield y AWS WAF permiten la identificación temprana de riesgos comunes de seguridad y la adecuación a los riesgos mencionados por OWASP. |
| **Medición** | Cantidad de ataques detectados.  Desempeño del sitio durante un ataque. |

|  |  |
| --- | --- |
| **Escenario** | Consulta y Modificación de Reglas de Negocio |
| **Justificación** | El funcionamiento de cuentas bancarias requiere la definición de reglas de negocio de forma dinámica, sin afectar el funcionamiento del sitio. |
| **Atributos de Calidad** | Capacidad de ser modificado, Disponibilidad |
| **Estímulos** | Necesidad de consultar reglas de negocio. |
| **Respuesta** | Resultados de la ejecución de las reglas de negocio. |
| **Decisiones de Arquitectura** | Utilizar un servidor Drools dentro de una instancia de AWS, con acceso tipo REST dentro de la infraestructura. |
| **Medición** | Tiempo de consulta al servidor de reglas. |

|  |  |
| --- | --- |
| **Escenario** | Orquestación de Servicios |
| **Justificación** | Al estar definido bajo una infraestructura SOA es necesario tener definido un mecanismo concreto de orquestación entre los servicios. |
| **Atributos de Calidad** | Disponibilidad, interoperabilidad |
| **Estímulos** | Necesidad de comunicación entre los servicios. |
| **Respuesta** | Comunicación eficiente entre los servicios. |
| **Decisiones de Arquitectura** | Utilizar protocolos de alto desempeño dentro de la arquitectura. Para no atar la arquitectura a un proveedor se tienen canales redundantes: el primer canal redundante es el uso de AWS SQS, y un canal adicional basado en protocolo AMQP y RabbitMQ. |
| **Medición** | Tiempo de respuesta de la comunicación entre servicios.  Cantidad de peticiones procesadas por SQS  Cantidad de peticiones procesadas por RabbitMQ |

|  |  |
| --- | --- |
| **Escenario** | Verificación de Clientes |
| **Justificación** | Como una plataforma de comercio electrónico con uso de tarjetas de crédito es necesario tener un mecanismo estándar para medir el riesgo financiero de los usuarios. |
| **Atributos de Calidad** | Corrección funcional, confidencialidad, no repudio |
| **Estímulos** | Pagos dentro de la plataforma |
| **Respuesta** | Resultado de la operación de pago. |
| **Decisiones de Arquitectura** | Se decidió utilizar una integración con centrales de riesgos, de forma que la información provista por la central permita la verificación del riesgo del usuario. |
| **Medición** | Tiempo de proceso de pagos.  Cantidad de transacciones en la plataforma.  Tiempo de respuesta de las centrales de riesgos.  Resultado de pago del cliente vs riesgo reportado (análisis de fiabilidad de la central de riesgo) |

1. Vista de casos de uso