CONSTRUCCIÓN DE PAQUETES CASO PRACTICO

CONSTRUCTION OF PACKAGES CASE STUDY

Said Antonio Assam Zaragoza
Salvador Alejandro Carrillo Ramírez
Eduardo Valentín Hernández Aguilar

INTRODUCCIÓN

En el siguiente trabajo pretendemos presentar una serie de concepto y definiciones ejercicios y acciones propios del estudio de construcción de paquetes.

La construcción de paquetes ayuda al desarrollo de proyectos que permitan facilitar funciones acciones o procesos según sea el caso que requiere o solicite el cliente.

Cabe resaltar que conforme a nuestro trabajo investigamos, también en nuestra región centro-occidente de México con respecto a los avances, las tendencias y las dirección o rumbos asía donde se dirige el mercado que es lo que se está demandando, la oportunidad que los estudiantes y los ya profesionistas podemos ejercer en la actualidad es de mucha importancia resaltar que el hecho de poder hacer trabajo desde casa esto amplio en demasía nuestro campo laboral, el estar en sintonía con esta rama de la informática nos proporciona un gran nicho de oportunidades, para ejercer lo relacionado a la construcción de paquetes.

RESUMEN

Los sistemas de información nacen en necesidad de las organizaciones de manejar grandes volúmenes de datos de una forma rápida, ágil y efectiva para su posterior intercambio, con el fin de apoyar las actividades de la empresa, negocio o Institución Educativa.

De este mismo modo podemos resaltar que la construcción de paquetes suma día con día al desarrollo de la sociedad permitiendo el alcance de productos o acciones que solo se veían como sueños futuristas, mismo que se catalogados como difíciles de poder ni siquiera coincidir la idea de cómo sería capaz de que simple paquetes dieran como resultado los grandes logros con los que actualmente podemos contar como por ejemplo manipular o tripular algún dispositivo o artefacto a distancia y con la presión del tiempo real, acciones que modificaran el futuro de nuestro mundo.

ABSTRACT

Information systems are born out of the need for organizations to handle large volumes of data in a fast, agile and effective way for their subsequent exchange, in order to support the activities of the company, business or Educational Institution.

In this same way, we can highlight that the construction of packages adds day by day to the development of society, allowing the scope of products or actions that were only seen as futuristic dreams, even though they are classified as difficult to be able to even agree on the idea of how It would be capable of simple packages resulting in the great achievements that we can currently count on, such as manipulating or manning a device or artifact at a distance and with the pressure of real time, actions that would modify the future of our world.

¿QUÉ ES LA CP?

El término se refiere a cierto software de aplicación diseñado para atender necesidades sectoriales, de un tipo de negocio, etc.

Un paquete integrado contiene un conjunto de programas para atender diversas necesidades, por ejemplo: contabilidad, ventas, etiquetas, etc. Con frecuencia, un paquete integra aplicaciones desarrolladas por distintas firmas.

Y se caracterizan por:

- Tienen un Interfaz con el que nos sentimos más cómodos trabajando.

- Se crean de manera que las aplicaciones pueden intercambiar ficheros sin dificultad y sin pérdida de información.

- Se pueden generar archivos en una aplicación que será insertado dentro de otra y que puede ser modificado con la aplicación que lo creó. Algunas de las herramientas son compartidas por todas las aplicaciones.

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

¿Qué pasa en nuestra región centro-occidente de México con respecto a la CP, Cuantos o qué elementos en los CP se utilizan?

Cabe señalar que las organizaciones no gubernamentales fueron las primeras en aprovechar las ventajas de Internet. Tal fue el caso del Ejército Zapatista de Liberación Nacional (EZLN), la cual fincó una sólida presencia en Internet desde la primavera de 1994.

En ese mismo periodo, instituciones académicas, periodísticas o de grupos sociales también establecieron sus propios servidores de información. Tal hecho resultó positivo, ya que dio pie a la creación de los primeros centros de servicio informativo en línea para la sociedad.

Desde entonces y hasta la fecha Internet se ha convertido en elemento clave en el nuevo entorno mundial. Por tal razón, para apoyar a las nuevas generaciones ante un contexto cada vez más digitalizado resulta necesario el desarrollo de nuevas competencias. Al sistema educativo le toca realizar cambios paralelos a los del entorno para cumplir satisfactoriamente con su compromiso ante la sociedad, así como las empresas y emprendedores que buscas obtener la mayor cantidad de desarrollos con los que se pueda cambiar a la región donde viven al país y a su vez el mundo hay demasiada inversión para este rubro tanto público como privado.

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Quién hace qué, ¿dónde, ¿cómo ¿en cuánto la CP? Ejemplifique.

Un proyecto de construcción de paquetes puede referirse, por ejemplo, al desarrollo de nuevos programas informáticos o a la aplicación de una solución de sistema de información, como una mejora técnica para cambiar los procesos de una organización. Las prioridades de un proyecto informático Para dirigir un proyecto informático, el/la Director·a del proyecto se apoya en datos cualitativos y cuantitativos, en pro de satisfacer las expectativas a diversos niveles:

Costes: el proyecto debe respetar el presupuesto definido al inicio del mismo. El control de los costes se logra anticipando los riesgos de desviación y aplicando medidas correctivas para no exceder el presupuesto asignado.

Plazos: el proyecto debe llevarse a cabo dentro de los plazos anunciados al cliente, siguiendo hitos intermedios. El respeto de los plazos implica la identificación de posibles desviaciones del calendario inicial y la aplicación de medidas correctivas.

Calidad: la ejecución del proyecto también implica fases de validación con las diversas partes involucradas. La garantía de calidad se logra comprobando el cumplimiento de los requisitos de:

ANÁLISIS: conformidad con las especificaciones de la aplicación,

**DISEÑO:** conformidad con las necesidades del cliente.

PRODUCTO FINAL: conformidad con las especificaciones establecidas aguas arriba.

Partes interesadas en un proyecto informático

En la realización de cualquier proyecto informático, participan diferentes actores:

El contratante: es aquel que espera resultados concretos. Corresponde a este actor definir los objetivos, el presupuesto y los plazos.

El contratista: es el "proveedor" del proyecto; es decir, aquel que ejecuta el trabajo. A este actor le corresponde diseñar y proponer soluciones, realizar pruebas antes de la entrega y respetar los costes y plazos establecidos.

El/la Director·a del proyecto: es quien dirige el proyecto y asume la responsabilidad del mismo.
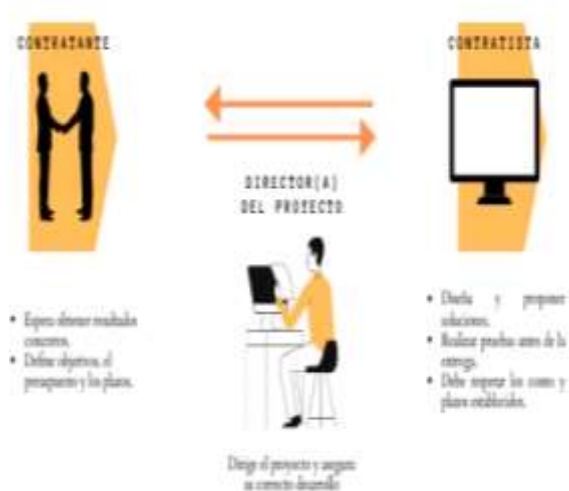
Que estrategias se realizan para implementar CP de un proceso o aplicativo en línea o vía móvil, servicios o atención al cliente, etc. con funciones propias y de paquetería. Ejemplifique con datos propios.

Una metodología es un conjunto de métodos utilizados para la investigación científica, en este paradigma, la metodología es la que permite diseñar nuestra estrategia en la que nos apoyamos en herramientas como la fueron:

Enterprise. Una herramienta ERP/CRM específicamente desarrollada para pequeñas y medianas empresas de distribución y/o fabricación. De instalación sencilla y altamente versátil, ofreciendo más de 40 soluciones sectoriales.

valentina estudio. Herramienta de base de datos que nos permitió vincular nuestro proyecto Cree, administre, consulte y explore bases de datos Valentina DB, MySQL, MariaDB, PostgreSQL y SQLite GRATIS.

Y de esta forma poder crear nuestra página

http://jocypizzas.com/

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Realiza el esquema de tu CP de pasar de código propio o de un aplicativo para por ejemplo la Generación de paquetería / analizar el estado de seguridad o integridad de almacenamiento de un DBMS personal de una PYME.

Model

Package in package ''

Functional Requirements Analysis

Package in package 'Model'

Functional requirements analysis examines the requirements for a solution that describe the capabilities that a solution must have in terms of the behaviour and information that the solution will manage. *BABOK® Guide v3*

Functional Requirements Hierarchy diagram

Custom diagram in package 'Functional Requirements Analysis'

This BABOK Requirements diagram shows three requirements that form a hierarchy showing how requirements can be broken down to any number of levels. Requirements can also be nested under each other in the Project Browser showing their hierarchy.
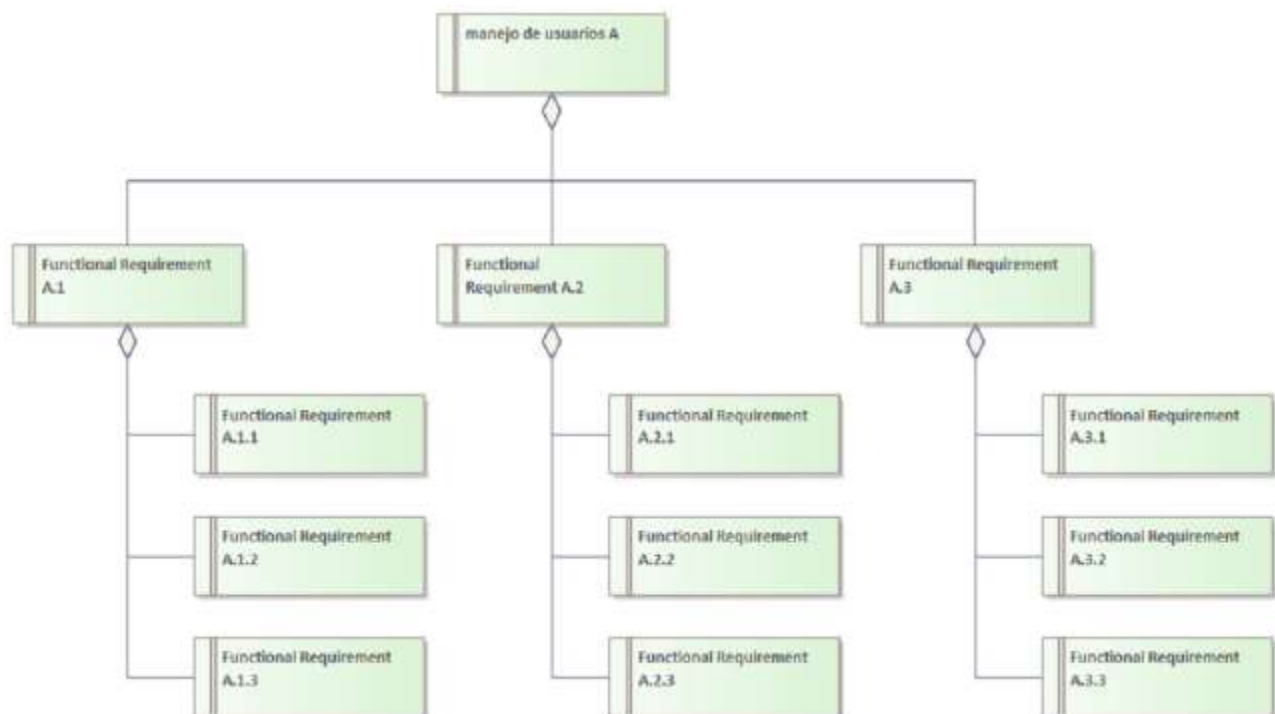
Figure 1: Functional Requirements Hierarchy

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Functional Requirement A.2

Functional Requirement «functional requirement» in package 'Functional Requirements Analysis'

NTS OWNED BY Functional Requirement A.2
tional Requirement A.2.1 : Functional Requirement «functional requirement»
tional Requirement A.2.2 : Functional Requirement «functional requirement»
tional Requirement A.2.3 : Functional Requirement «functional requirement»

OING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.2 to «functional requirement» manejo de usuarios A [Direction is 'Source -> Destination'.]

MING STRUCTURAL RELATIONSHIPS
egation   from «functional requirement» Functional Requirement A.2.3 to «functional requirement» Functional Requirement A.2 [Direction is 'Source -> Destination'.]
egation   from «functional requirement» Functional Requirement A.2.1 to «functional requirement» Functional Requirement A.2 [Direction is 'Source -> Destination'.]
egation   from «functional requirement» Functional Requirement A.2.2 to «functional requirement» Functional Requirement A.2 [Direction is 'Source -> Destination'.]

Functional Requirement A.2.1

Functional Requirement «functional requirement» owned by 'Functional Requirement A.2', in package 'Functional Requirements Analysis'

OING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.2.1 to «functional requirement» Functional Requirement A.2 [Direction is 'Source -> Destination'.]

Functional Requirement A.2.2

Functional Requirement «functional requirement» owned by 'Functional Requirement A.2', in package 'Functional Requirements Analysis'

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022

egation from «functional requirement» Functional Requirement A.2.2 to «functional requirement» Functional Requirement A.2

[Direction is 'Source -> Destination'.]

Functional Requirement A.2.3

Functional Requirement «functional requirement» owned by 'Functional Requirement A.2', in package 'Functional Requirements Analysis'

Functional Requirement A.2.3
Version 1.0 Phase 1.0 Validated
saidy created on 28/01/2023. Last modified 28/01/2023

DING STRUCTURAL RELATIONSHIPS

egation from «functional requirement» Functional Requirement A.2.3 to «functional requirement» Functional Requirement A.2

[Direction is 'Source -> Destination'.]

Functional Requirement A.3

Functional Requirement «functional requirement» in package 'Functional Requirements Analysis'

Functional Requirement A.3
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

NTS OWNED BY Functional Requirement A.3

tional Requirement A.3.1 : Functional Requirement «functional requirement»

tional Requirement A.3.2 : Functional Requirement «functional requirement»

tional Requirement A.3.3 : Functional Requirement «functional requirement»

DING STRUCTURAL RELATIONSHIPS

egation from «functional requirement» Functional Requirement A.3 to «functional requirement» manejo de usuarios A

[Direction is 'Source -> Destination'.]

MING STRUCTURAL RELATIONSHIPS

egation from «functional requirement» Functional Requirement A.3.2 to «functional requirement» Functional Requirement A.3

[Direction is 'Source -> Destination'.]

egation from «functional requirement» Functional Requirement A.3.3 to «functional requirement» Functional Requirement A.3

[Direction is 'Source -> Destination'.]

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022

MING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.3.1 to «functional requirement» Functional Requirement A.3
[Direction is 'Source -> Destination'.]

Functional Requirement A.3.1

Functional Requirement «functional requirement» owned by 'Functional Requirement A.3', in package 'Functional Requirements Analysis'

Functional Requirement A.3.1
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

DING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.3.1 to «functional requirement» Functional Requirement A.3
[Direction is 'Source -> Destination'.]

Functional Requirement A.3.2

Functional Requirement «functional requirement» owned by 'Functional Requirement A.3', in package 'Functional Requirements Analysis'

Functional Requirement A.3.2
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

DING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.3.2 to «functional requirement» Functional Requirement A.3
[Direction is 'Source -> Destination'.]

Functional Requirement A.3.3

Functional Requirement «functional requirement» owned by 'Functional Requirement A.3', in package 'Functional Requirements Analysis'

Functional Requirement A.3.3
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

DING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.3.3 to «functional requirement» Functional Requirement A.3
[Direction is 'Source -> Destination'.]

manejo de usuarios A

Functional Requirement «functional requirement» in package 'Functional Requirements Analysis'

manejo de usuarios A
Version 1.0 Phase 1.0 Approved
saidy created on 28/01/2023. Last modified 28/01/2023
Alias roles
Keywords id, password, sha512

NTS OWNED BY manejo de usuarios A
:tional Requirement A.1 : Functional Requirement «functional requirement»
:tional Requirement A.1.2 : Functional Requirement «functional requirement»
:tional Requirement A.1.3 : Functional Requirement «functional requirement»

MING STRUCTURAL RELATIONSHIPS
egation   from «functional requirement» Functional Requirement A.3 to «functional requirement» manejo de usuarios A [Direction is 'Source -> Destination'.]
egation   from «functional requirement» Functional Requirement A.1 to «functional requirement» manejo de usuarios A [Direction is 'Source -> Destination'.]
egation   from «functional requirement» Functional Requirement A.2 to «functional requirement» manejo de usuarios A [Direction is 'Source -> Destination'.]

Functional Requirement A.1

Functional Requirement «functional requirement» owned by 'manejo de usuarios A', in package 'Functional Requirements Analysis'

Functional Requirement A.1
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

NTS OWNED BY Functional Requirement A.1
:tional Requirement A.1.1 : Functional Requirement «functional requirement»

OING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.1 to «functional requirement» manejo de usuarios A [Direction is 'Source -> Destination'.]

MING STRUCTURAL RELATIONSHIPS
egation   from «functional requirement» Functional Requirement A.1.3 to «functional requirement» Functional Requirement A.1 [Direction is 'Source -> Destination'.]
egation   from «functional requirement» Functional Requirement A.1.1 to «functional requirement» Functional Requirement A.1 [Direction is 'Source -> Destination'.]

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022

MING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.1.2 to «functional requirement» Functional Requirement A.1 [Direction is 'Source -> Destination'.]

Functional Requirement A.1.1

Functional Requirement «functional requirement» owned by 'Functional Requirement A.1', in package 'Functional Requirements Analysis'

Functional Requirement A.1.1
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

DING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.1.1 to «functional requirement» Functional Requirement A.1 [Direction is 'Source -> Destination'.]

Functional Requirement A.1.2

Functional Requirement «functional requirement» owned by 'manejo de usuarios A', in package 'Functional Requirements Analysis'

Functional Requirement A.1.2
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

DING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.1.2 to «functional requirement» Functional Requirement A.1 [Direction is 'Source -> Destination'.]

Functional Requirement A.1.3

Functional Requirement «functional requirement» owned by 'manejo de usuarios A', in package 'Functional Requirements Analysis'

Functional Requirement A.1.3
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

DING STRUCTURAL RELATIONSHIPS
egation from «functional requirement» Functional Requirement A.1.3 to «functional requirement» Functional Requirement A.1 [Direction is 'Source -> Destination'.]

No funcional A plataform

Package in package 'Model'

Non-functional requirements analysis examines the requirements for a solution that define how well the functional requirements must perform. It specifies criteria that can be used to judge the operation of a system rather than specific behaviours (which are referred to as the functional requirements). *BABOK® Guide v3*

no funcional A plataform
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

DBMS diagram

Custom diagram in package 'no funcional A plataform'

This Package diagram contains the packages that collect the non-functional requirements into groups. Other packages can be added as they are completed. It provides a convenient overview of the non-Functional Requirements and their groups. The Requirements can be removed from this diagram.

Availability
- ☑ + Availability Requirement A
- ☑ + Availability Requirement B

The degree to which the solution is operable and accessible when required for use, often expressed in terms of percent of time the solution is available.

Performance Efficiency

The degree to which a solution or component performs its designated functions with minimum consumption of resources. Can be defined based on the context or period, such as high-peak, midpeak or off-peak usage.

Usability

The ease with which a user can learn to use the solution.

Compatibility
- ☑ + Compatibility Requirement C

The degree to which the solution operates effectively with other components in its environment, such as one process with another.

Portability

The ease with which a solution or component can be transferred from one environment to another.

Certification

The constraints on the solution that are necessary to meet certain standards or industry conventions.

Extensibility

The ability of a solution to incorporate new functionality.

Reliability

The ability of a solution or component to perform its required functions under stated conditions for a specified period of time, such as mean time to failure of a device.

Compliance

The regulatory, financial, or legal constraints which can vary based on the context or jurisdiction.

Maintainability

The ease with which a solution or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

Scalability

The degree with which a solution is able to grow or evolve to handle increased amounts of work.

Localization

The requirements dealing with local languages, laws, currencies, cultures, spellings, and other characteristics of users, which requires attention to the context.

Functionality

The degree to which the solution functions meet user needs, including aspects of suitability, accuracy, and interoperability.

Security

The aspects of a solution that protect solution content or solution components from accidental or malicious access, use, modification, destruction, or disclosure.

Service Level Agreements

The constraints of the organization being served by the solution that are formally agreed to by both the provider and the user of the solution.

Figure 2: DBMS

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Availability

Package in package 'no funcional A plataform'

The degree to which the solution is operable and accessible when required for use, often expressed in terms of percent of time the solution is available.

<div align="right">

Availability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Compatibility

Package in package 'no funcional A plataform'

The degree to which the solution operates effectively with other components in its environment, such as one process with another.

<div align="right">

Compatibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Extensibility

Package in package 'no funcional A plataform'

The ability of a solution to incorporate new functionality.

<div align="right">

Extensibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Maintainability

Package in package 'no funcional A plataform'

The ease with which a solution or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

<div align="right">

Maintainability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Functionality

Package in package 'no funcional A plataform'

The degree to which the solution functions meet user needs, including aspects of suitability, accuracy, and interoperability.

<div align="right">

Functionality
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Performance Efficiency

Package in package 'no funcional A plataform'

The degree to which a solution or component performs its designated functions with minimum consumption of resources. Can be defined based on the context or period, such as high-peak, midpeak or off-peak usage.

<div align="right">

Performance Efficiency
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Portability

Package in package 'no funcional A plataform'

The ease with which a solution or component can be transferred from one environment to another.

<div align="right">

Portability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Reliability

Package in package 'no funcional A plataform'

The ability of a solution or component to perform its required functions under stated conditions for a specified period of time, such as mean time to failure of a device.

<div align="right">

Reliability

</div>

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Scalability
Package in package 'no funcional A plataform'
The degree with which a solution is able to grow or evolve to handle increased amounts of work.

Security
Package in package 'no funcional A plataform'
The aspects of a solution that protect solution content or solution components from accidental or malicious access, use, modification, destruction, or disclosure.

Usability
Package in package 'no funcional A plataform'
The ease with which a user can learn to use the solution.

Certification
Package in package 'no funcional A plataform'
The constraints on the solution that are necessary to meet certain standards or industry conventions.

Compliance
Package in package 'no funcional A plataform'
The regulatory, financial, or legal constraints which can vary based on the context or jurisdiction.

Localization
Package in package 'no funcional A plataform'
The requirements dealing with local languages, laws, currencies, cultures, spellings, and other characteristics of users, which requires attention to the context.

Service Level Agreements
Package in package 'no funcional A plataform'
The constraints of the organization being served by the solution that are formally agreed to by both the provider and the user of the solution.

sistema operativo diagram

Custom diagram in package 'no funcional A plataform'

This Package diagram contains the packages that collect the non-functional requirements into groups. Other packages can be added as they are completed. It provides a convenient overview of the non-Functional Requirements and their groups. The Requirements can be removed from this diagram.

Availability
- + Availability Requirement A
- + Availability Requirement B

The degree to which the solution is operable and accessible when required for use, often expressed in terms of percent of time the solution is available.

Performance Efficiency
The degree to which a solution or component performs its designated functions with minimum consumption of resources. Can be defined based on the context or period, such as high-peak, midpeak or off-peak usage.

Usability
The ease with which a user can learn to use the solution.

Compatibility
- + Compatibility Requirement C

The degree to which the solution operates effectively with other components in its environment, such as one process with another.

Portability
The ease with which a solution or component can be transferred from one environment to another.

Certification
The constraints on the solution that are necessary to meet certain standards or industry conventions.

Extensibility
The ability of a solution to incorporate new functionality.

Reliability
The ability of a solution or component to perform its required functions under stated conditions for a specified period of time, such as mean time to failure of a device.

Compliance
The regulatory, financial, or legal constraints which can vary based on the context or jurisdiction.

Maintainability
The ease with which a solution or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

Scalability
The degree with which a solution is able to grow or evolve to handle increased amounts of work.

Localization
The requirements dealing with local languages, laws, currencies, cultures, spellings, and other characteristics of users, which requires attention to the context.

Functionality
The degree to which the solution functions meet user needs,including aspects of suitability, accuracy, and interoperability.

Security
The aspects of a solution that protect solution content or solution components from accidental or malicious access, use, modification, destruction, or disclosure.

Service Level Agreements
The constraints of the organization being served by the solution that are formally agreed to by both the provider and the user of the solution.

Figure 3: sistema operativo

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Availability

Package in package 'no funcional A plataform'

The degree to which the solution is operable and accessible when required for use, often expressed in terms of percent of time the solution is available.

Availability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Compatibility

Package in package 'no funcional A plataform'
The degree to which the solution operates effectively with other components in its environment, such as one process with another.

Compatibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Extensibility

Package in package 'no funcional A plataform'
The ability of a solution to incorporate new functionality.

Extensibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Maintainability

Package in package 'no funcional A plataform'
The ease with which a solution or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

Maintainability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Functionality

Package in package 'no funcional A plataform'
The degree to which the solution functions meet user needs, including aspects of suitability, accuracy, and interoperability.

Functionality
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Performance Efficiency

Package in package 'no funcional A plataform'
The degree to which a solution or component performs its designated functions with minimum consumption of resources. Can be defined based on the context or period, such as high-peak, midpeak or off-peak usage.

Performance Efficiency
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Portability

Package in package 'no funcional A plataform'
The ease with which a solution or component can be transferred from one environment to another.

Portability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Reliability

Package in package 'no funcional A plataform'
The ability of a solution or component to perform its required functions under stated conditions for a specified period of time, such as mean time to failure of a device.

Scalability

Package in package 'no funcional A plataform'

The degree with which a solution is able to grow or evolve to handle increased amounts of work.

<div align="right">

Scalability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Security

Package in package 'no funcional A plataform'

The aspects of a solution that protect solution content or solution components from accidental or malicious access, use, modification, destruction, or disclosure.

<div align="right">

Security
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Usability

Package in package 'no funcional A plataform'

The ease with which a user can learn to use the solution.

<div align="right">

Usability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Certification

Package in package 'no funcional A plataform'

The constraints on the solution that are necessary to meet certain standards or industry conventions.

<div align="right">

Certification
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Compliance

Package in package 'no funcional A plataform'

The regulatory, financial, or legal constraints which can vary based on the context or jurisdiction.

<div align="right">

Compliance
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Localization

Package in package 'no funcional A plataform'

The requirements dealing with local languages, laws, currencies, cultures, spellings, and other characteristics of users, which requires attention to the context.

<div align="right">

Localization
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Service Level Agreements

Package in package 'no funcional A plataform'

The constraints of the organization being served by the solution that are formally agreed to by both the provider and the user of the solution.

<div align="right">

Service Level Agreements
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Availability

Package in package 'no funcional A plataform'

The degree to which the solution is operable and accessible when required for use, often expressed in terms of percent of time the solution is available. Availability

<div align="right">

Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Availability Requirement A

NonfunctionalRequirement «NonfunctionalRequirement» in package 'Availability'

<div align="right">

Availability Requirement A
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Availability Requirement B

NonfunctionalRequirement «NonfunctionalRequirement» in package 'Availability'

<div align="right">

Availability Requirement B
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Compatibility

Package in package 'no funcional a plataform'

The degree to which the solution operates effectively with other components in its environment, such as one process with another.

<div align="right">

Compatibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Compatibility Requirement C

NonfunctionalRequirement «NonfunctionalRequirement» in package 'Compatibility'

<div align="right">

Compatibility Requirement C
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Extensibility

Package in package 'no funcional A plataform'

The ability of a solution to incorporate new functionality.

<div align="right">

Extensibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Maintainability

Package in package 'no funcional A plataform'

The ease with which a solution or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

<div align="right">

Maintainability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Functionality

Package in package 'no funcional A plataform'

The degree to which the solution functions meet user needs, including aspects of suitability, accuracy, and interoperability.

<div align="right">

Functionality
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Performance Efficiency

Package in package 'no funcional A plataform'

The degree to which a solution or component performs its designated functions with minimum consumption of resources. Can be defined based on the context or period, such as high-peak, midpeak or off-peak usage.

<div align="right">

Performance Efficiency
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023
</div>

Portability

Package in package 'no funcional A plataform'

The ease with which a solution or component can be transferred from one environment to another.

Reliability

Package in package 'no funcional A plataform'

The ability of a solution or component to perform its required functions under stated conditions for a specified period of time, such as mean time to failure of a device.

Reliability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Scalability

Package in package 'no funcional A plataform'

The degree with which a solution is able to grow or evolve to handle increased amounts of work.

Scalability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Security

Package in package 'no funcional A plataform'

The aspects of a solution that protect solution content or solution components from accidental or malicious access, use, modification, destruction, or disclosure.

Security
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Usability

Package in package 'no funcional A plataform'

The ease with which a user can learn to use the solution.

Usability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Certification

Package in package 'no funcional A plataform'

The constraints on the solution that are necessary to meet certain standards or industry conventions.

Certification
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Compliance

Package in package 'no funcional A plataform'

The regulatory, financial, or legal constraints which can vary based on the context or jurisdiction.

Compliance
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Localization

Package in package 'no funcional A plataform'

The requirements dealing with local languages, laws, currencies, cultures, spellings, and other characteristics of users, which requires attention to the context.

Localization
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Service Level Agreements

Package in package 'no funcional A plataform'

The constraints of the organization being served by the solution that are formally agreed to by both the provider and the user of the solution.

Service Level Agreements
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Non-Functional - Fluidez

Package in package 'Model'

Non-functional requirements analysis examines the requirements for a solution that define how well the functional requirements must perform. It specifies criteria that can be used to judge the operation of a system rather than specific behaviours (which are referred to as the functional requirements). *BABOK® Guide v3*

<div align="right">

Non-Functional - Fluidez
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Non-Functional Requirements Analysis diagram

Custom diagram in package 'Non-Functional - Fluidez'

This Package diagram contains the packages that collect the non-functional requirements into groups. Other packages can be added as they are completed. It provides a convenient overview of the non-Functional Requirements and their groups. The Requirements can be removed from this diagram.

<div align="right">

Non-Functional Requirements Analysis
Version 1.0
saidy created on 28/01/2023.   Last modified 28/01/2023

</div>

Figure 4: Non-Functional Requirements Analysis

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022

Availability

Package in package 'Non-Functional - Fluidez'
The degree to which the solution is operable and accessible when required for use, often expressed in terms of percent of time the solution is available.

<div align="right">

Availability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Compatibility

Package in package 'Non-Functional - Fluidez'
The degree to which the solution operates effectively with other components in its environment, such as one process with another.

<div align="right">

Compatibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/202

</div>

Extensibility

Package in package 'Non-Functional - Fluidez'
The ability of a solution to incorporate new functionality.

<div align="right">

Extensibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Maintainability

Package in package 'Non-Functional - Fluidez'
The ease with which a solution or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

<div align="right">

Maintainability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Functionality

Package in package 'Non-Functional - Fluidez'
The degree to which the solution functions meet user needs, including aspects of suitability, accuracy, and interoperability.

<div align="right">

Functionality
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Performance Efficiency

Package in package 'Non-Functional - Fluidez'
The degree to which a solution or component performs its designated functions with minimum consumption of resources. Can be defined based on the context or period, such as high-peak, midpeak or off-peak usage.

<div align="right">

Performance Efficiency
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Portability

Package in package 'Non-Functional - Fluidez'
The ease with which a solution or component can be transferred from one environment to another.

<div align="right">

Portability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Reliability

Package in package 'Non-Functional - Fluidez'
The ability of a solution or component to perform its required functions under stated conditions for a specified period of time, such as mean time to failure of a device.

<div align="right">

Reliability
Version 1.0 Phase 1.0 Proposed

</div>

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Scalability

Package in package 'Non-Functional - Fluidez'

The degree with which a solution is able to grow or evolve to handle increased amounts of work.

<div align="right">

Scalability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Security

Package in package 'Non-Functional - Fluidez'

The aspects of a solution that protect solution content or solution components from accidental or malicious access, use, modification, destruction, or disclosure.

<div align="right">

Security
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Usability

Package in package 'Non-Functional - Fluidez'

The ease with which a user can learn to use the solution.

<div align="right">

Usability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Certification

Package in package 'Non-Functional - Fluidez'

The constraints on the solution that are necessary to meet certain standards or industry conventions.

<div align="right">

Certification
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Compliance

Package in package 'Non-Functional - Fluidez'

The regulatory, financial, or legal constraints which can vary based on the context or jurisdiction

<div align="right">

Compliance
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Localization

Package in package 'Non-Functional - Fluidez'

The requirements dealing with local languages, laws, currencies, cultures, spellings, and other characteristics of users, which requires attention to the context.

<div align="right">

Localization
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Service Level Agreements

Package in package 'Non-Functional - Fluidez'

The constraints of the organization being served by the solution that are formally agreed to by both the provider and the user of the solution.

<div align="right">

Service Level Agreements
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Availability

Package in package 'Non-Functional - Fluidez'

The degree to which the solution is operable and accessible when required for use, often expressed in terms of percent of time the solution is available.

<div align="right">

Availability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023.   Last modified 28/01/2023
</div>

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022

Availability Requirement A
NonfunctionalRequirement «NonfunctionalRequirement» in package 'Availability'

<div align="right">

Availability Requirement A
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Availability Requirement B
NonfunctionalRequirement «NonfunctionalRequirement» in package 'Availability'

<div align="right">

Availability Requirement B
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Compatibility
Package in package 'Non-Functional - Fluidez'
The degree to which the solution operates effectively with other components in its environment, such as one process with another.

<div align="right">

Compatibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Compatibility Requirement C
NonfunctionalRequirement «NonfunctionalRequirement» in package 'Compatibility'

<div align="right">

Compatibility Requirement C
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Extensibility
Package in package 'Non-Functional - Fluidez'
The ability of a solution to incorporate new functionality.

<div align="right">

Extensibility
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Maintainability
Package in package 'Non-Functional - Fluidez'
The ease with which a solution or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

<div align="right">

Maintainability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Functionality
Package in package 'Non-Functional - Fluidez'
The degree to which the solution functions meet user needs, including aspects of suitability, accuracy, and interoperability.

<div align="right">

Functionality
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Performance Efficiency
Package in package 'Non-Functional - Fluidez'
The degree to which a solution or component performs its designated functions with minimum consumption of resources. Can be defined based on the context or period, such as high-peak, midpeak or off-peak usage.

<div align="right">

Performance Efficiency
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

</div>

Portability
Package in package 'Non-Functional - Fluidez'
The ease with which a solution or component can be transferred from one environment to another.

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Reliability

Package in package 'Non-Functional - Fluidez'

The ability of a solution or component to perform its required functions under stated conditions for a specified period of time, such as mean time to failure of a device.

Reliability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Scalability

Package in package 'Non-Functional - Fluidez'

The degree with which a solution is able to grow or evolve to handle increased amounts of work.

Scalability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Security

Package in package 'Non-Functional - Fluidez'

The aspects of a solution that protect solution content or solution components from accidental or malicious access, use, modification, destruction, or disclosure.

Security
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Usability

Package in package 'Non-Functional - Fluidez'

The ease with which a user can learn to use the solution.

Usability
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Certification

Package in package 'Non-Functional - Fluidez'

The constraints on the solution that are necessary to meet certain standards or industry conventions.

Certification
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Compliance

Package in package 'Non-Functional - Fluidez'

The regulatory, financial, or legal constraints which can vary based on the context or jurisdiction.

Compliance
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Localization

Package in package 'Non-Functional - Fluidez'

The requirements dealing with local languages, laws, currencies, cultures, spellings, and other characteristics of users, which requires attention to the context.

Localization
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Service Level Agreements

Package in package 'Non-Functional - Fluidez'

The constraints of the organization being served by the solution that are formally agreed to by both the provider and the user of the solution.
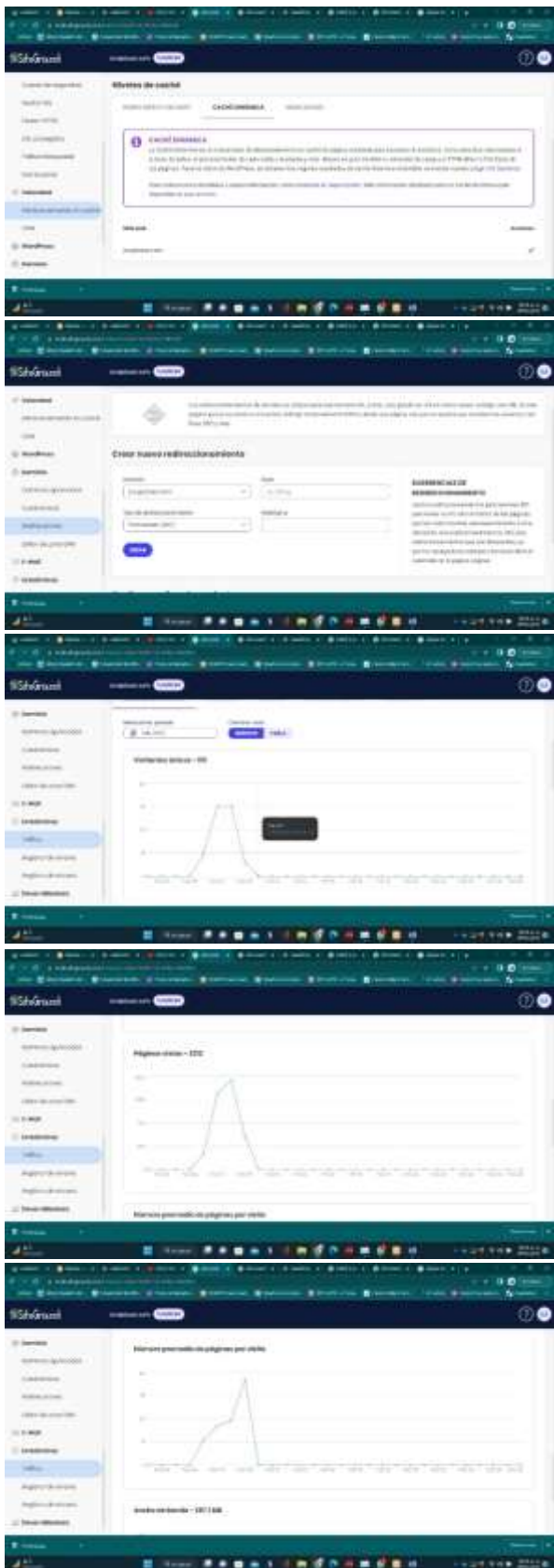
Service Level Agreements
Version 1.0 Phase 1.0 Proposed
saidy created on 28/01/2023. Last modified 28/01/2023

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022
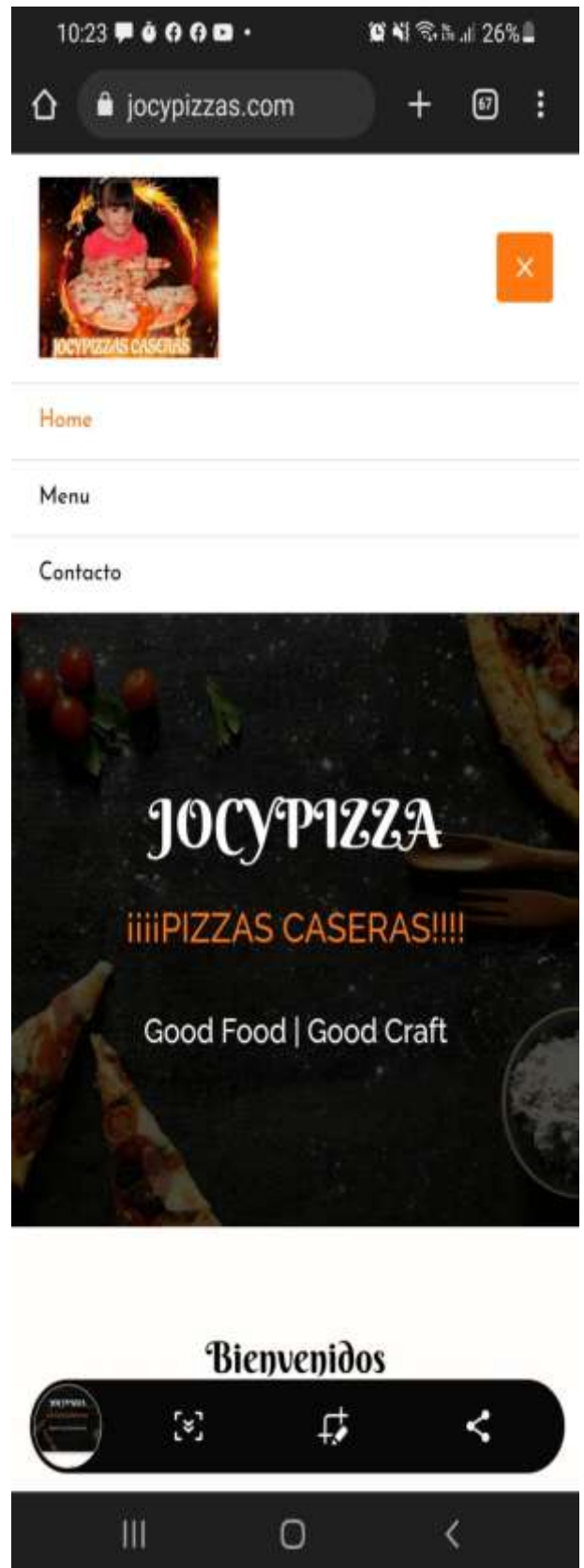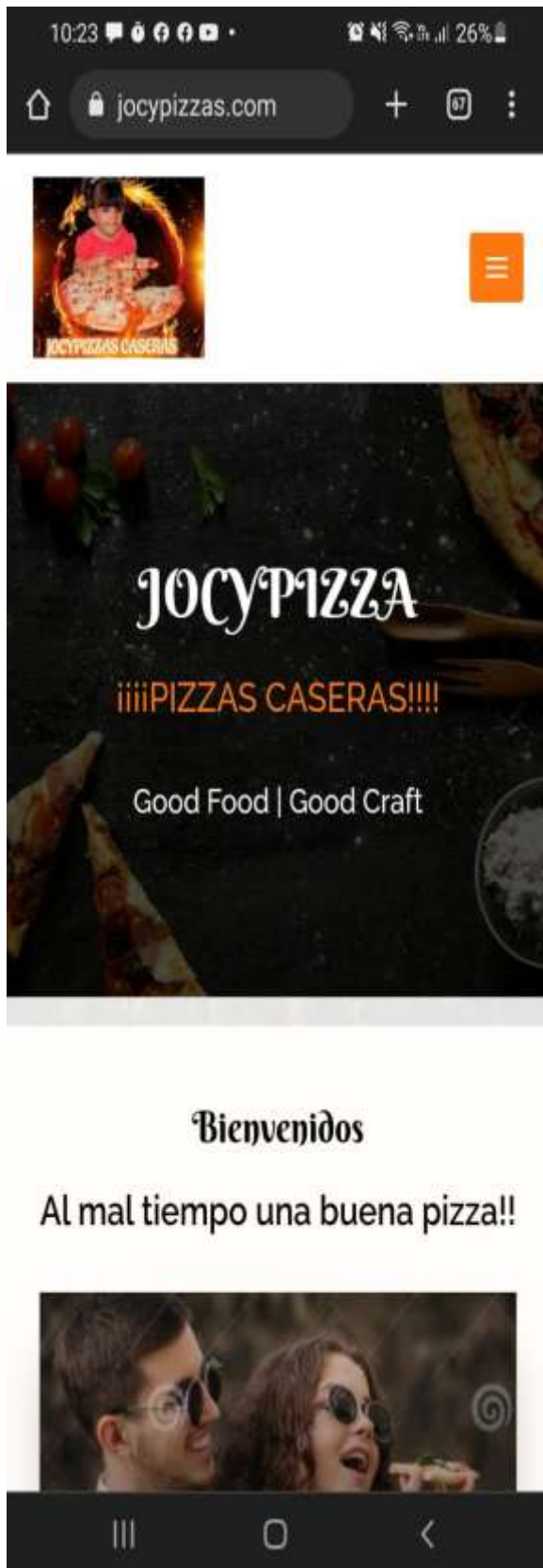
Integra fotos o video de los elementos de tu CP realizar una memoria fotográfica con datos de los elementos de la construcción de paquetes investigados por todos o el usuario en clases.
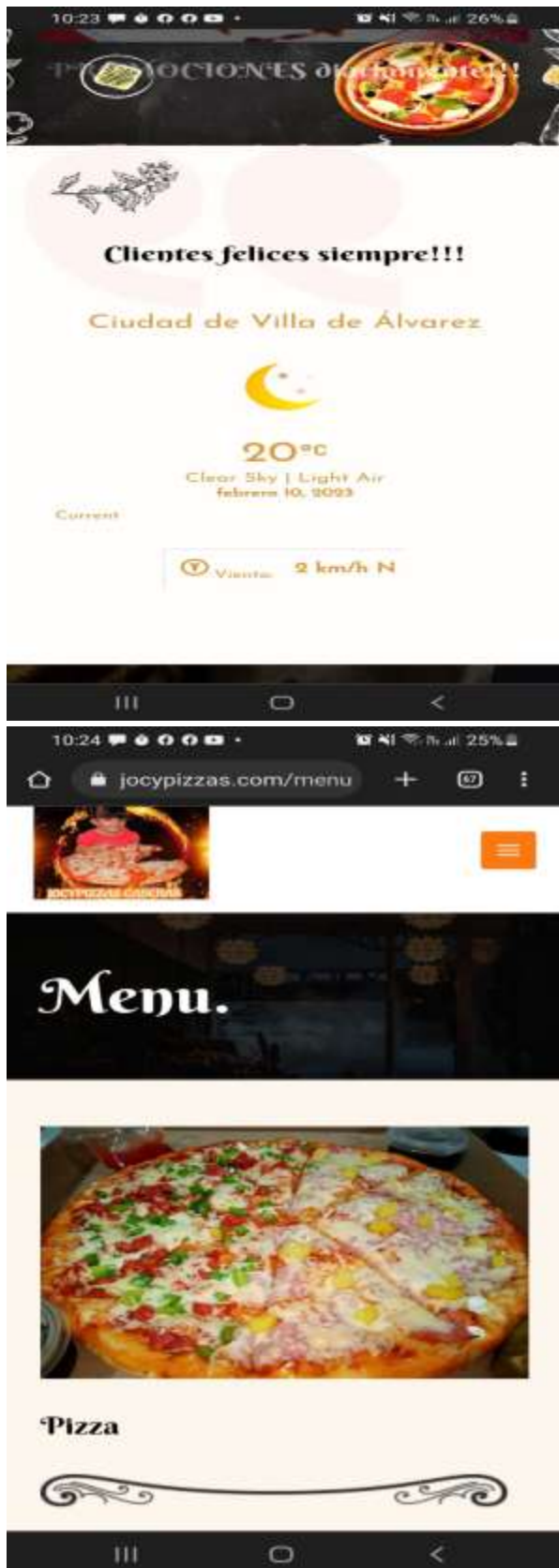

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022

PIZZAS TIENEN ORILLA
RELLENA DE QUESO
CHICA-MEDIANA-
GRANDE
$90.00-$130.00-$150.00

HAWAIANA

JAMON-PIÑA-EXTRA QUESO

MEXICANA

CHORIZO-PIMIENTO VERDE-
JAMON-JITOMATE-CEBOLLA-
EXTRAQUESO

CARNES FRIAS

PEPERONI-JAMON-SALAMI-EXTRA
QUESO

VEGETARIANA

CHAMPIÑON-PIMIENTO VERDE-
JITOMATE-CEBOLLA-ACEITUNA
NEGRA-PIÑA

PEPERONI ESPECIAL

PEPERONI-CHAMPIÑON-
EXTRAQUESO

Menu.

Pizza

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022

¡Sorprende me!!! ejemplificación o aportes de manera abierta con respecto a lo de CP

Hoy en día, no hay un proceso que pueda simplificarse aplicando una solución de software. Sin embargo, hay compañías y marcas que necesitan una solución de software mucho más avanzada que las que están disponibles para el público en general.

En pocas palabras, el desarrollo de software personalizado les da a las empresas y compañías la opción de tener una solución de software completamente hecha solo para ellos, una solución que se adapte directamente a sus necesidades y requisitos.

El desarrollo de software para clientes es una parte tan vital de las empresas en la actualidad, ya que se espera que el mercado de software empresarial alcance más de 405 mil millones de dólares estadounidenses para fin de año, según un estudio realizado por statista.

Teniendo esto en cuenta, echemos un vistazo a lo que realmente es el desarrollo de software personalizado y repasemos las cinco cosas principales que debe saber al respecto, incluida la forma en que puede ayudar a su empresa.

1. El primer programador en todo el mundo fue una mujer. Se llamaba Ada Lovelace y nació en 1815. Algunas personas creen que Lovelace publicó el primer algoritmo destinado a ser ejecutado por una máquina.

2. El primer lenguaje de programación del mundo se llamó FORTRAN (acrónimo de Formula Translation), y fue creado en 1956. El equipo que lo creó, dirigido por el Sr. John Backus, comenzó a trabajar en él en 1954.

3. Más del 70% de todos los trabajos de programación son en campos e industrias fuera de la tecnología. La gente que aprende a programar puede fortalecer innumerables habilidades del siglo XXI que le ayudarán a hacer un trabajo excepcional en casi todas las industrias.

4. El primer «bug de la computadora» fue llamado así a causa de un insecto real. Este bicho fue encontrado por Grace Hopper. El ordenador que ella estaba construyendo comenzó a fallar y fue entonces cuando descubrió una polilla de verdad en el sistema. Desde entonces, cuando algo falla en el software o en el

hardware, lo llamamos «bug». Si quieres leer más sobre la historia y el legado de Grace Hopper, dale una mirada a esta entrada que escribimos hace algunas semanas.

5. Hoy en día, ¡hay más de 700 lenguajes de programación en el mundo! Los niños pueden empezar a aprender a programar a través de sistemas amigables y coloridos como Scratch, para luego progresar a lenguajes de programación más avanzados y complejos como Java o Python.

6. Se necesitan menos códigos para enviar a un hombre a la luna que para manejar un celular.

7. Hablando de la luna… Margaret Hamilton, una informática e ingeniería de sistemas americana, escribió el código informático que ayudó a salvar la misión de aterrizaje del Apolo en la luna.

8. El primer juego de computadora no generó ninguna ganancia. Hoy en día la industria de los juegos vale 30 millones de dólares más que la industria del cine.

9. El primer virus informático no fue diseñado para ser dañino. ¿Qué pasó en el camino? ¡No tenemos ni idea!

10. Es un requisito indispensable que los astrónomos sepan programar. De hecho,

utilizan una variedad de lenguajes de programación para procesar las mediciones que hacen, y también para desarrollar simulaciones de fenómenos astrofísicos.

11. ¡Tus hijos pueden aprender a programar de una manera fácil, amigable, segura y accesible! A través de los cursos en vivo y en línea que brinda Tekkie Uni, tus hijos pueden tener clases con los mejores instructores disponibles y fortalecer habilidades extraordinarias mientras aprenden a programar sus propias aplicaciones,

juegos y software. La programación es para todos y está a la vuelta de la esquina.

¿Cuál de estos datos te sorprendió más? Comparte tus opiniones con nosotros en los comentarios. Si tienes algún otro dato divertido o sorprendente sobre la programación, no dudes en compartirlo con nosotros y podríamos añadirlo a esta lista. ¡Te agradecemos de manera anticipada!

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes**,** Colima, 2022

http://jocypizzas.com/

https://elinea.infraestructuragis.com/mod/assign/view.php?id=1107&forceview=1

https://www.google.com.mx/search?q=construccion+de+paquetes+de+software&hl=es&tbm=bks&sxsrf=AJOqlzXcD8lgckKfJ03nK1y-AKVVoh2byQ%3A1674264206316&ei=jj7LY8H-EvjHkPIPnLSmwAU&ved=0ahUKEwjB_ZuawNf8AhX4I0QIHRyaCVgQ4dUDCAk&uact=5&oq=construccion+de+paquetes+de+software&gs_lcp=Cg1nd3Mtd2l6LWJvb2tzEAM6BQgAEIAEOgQIIRAKUABY_kxgp09oBnAAeACAAZYBiAGtI5IBBDYuMzSYAQCgAQHAAQE&sclient=gws-wiz-books

https://www.google.com.mx/search?q=ingenieria+de+software&hl=es&tbm=bks&sxsrf=AJOqlzW0SssUhmrSOfQO7fvCKKpR5kShNA%3A1674264273998&ei=0T7LY4POPO

TLkPIPmMG68AY&oq=ingenieria+de+&gs_lcp=Cg1nd3Mtd2l6LWJvb2tzEAEYADIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgARQmw9Y-R1gwy1oAXAAeACAAXCIAfgMkgEEMTMuNJgBAKABAcABAQ&sclient=gws-wiz-books

 https://www.fergut.com/la-evolucion-de-internet-en-mexico-y-su-impacto-en-el-ambito-educativo-de-1986-a-2006/

https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/gestion-de-proyectos-informaticos-ejemplo

https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf

Autores: Said Antonio Assam Zaragoza, Salvador Alejandro Carrillo Ramírez, y Eduardo Valentín Hernández Aguilar Caso Práctico construcción de paquetes, Colima, 2022