

**Facultad de Ingeniería**

**Carrera Profesional de Ingeniería de Sistemas - Software**

**SISTEMA DE VENTAS DE JUGUETES PARA NIÑOS CON DISCAPACIDAD**

**INTEGRANTES**

**Córdova Facundo, Yalixa**

**Allcca Moriano, Miguel ángel**

**Aquino Gutierrez, John Jose**

**Ccispichito Huaman, Omar Paul**

**Yaurima Navarro, Alexander Maximo**

**PROFESOR**

**ROBLES FERNANDEZ, IVAN**

**CURSO**

**Curso Integrador I: Sistemas - Software**

**TURNO**

**Noche - Virtual**

**SECCIÓN**

**15739**

Índice

[**CAPÍTULO 1 ASPECTOS GENERALES** 8](#_Toc87647298)

[**1.1.** **Análisis de la empresa:** 8](#_Toc87647299)

[**Visión** 9](#_Toc87647300)

[**Misión** 9](#_Toc87647301)

[**Análisis del Entorno (PESTEL)** 9](#_Toc87647302)

[**Mapa de procesos**  11](#_Toc87647309)

[**Estrategia de Negocio (Tablero de Control de Mando)** 12](#_Toc87647310)

[**1.2. Descripción del Problema** 12](#_Toc87647311)

[**1.3.** **Objetivos** 13](#_Toc87647312)

[**1.3.1** **Objetivo General** 13](#_Toc87647313)

[**1.3.2** **Objetivos Específicos** 13](#_Toc87647314)

[**1.5.** **Justificación** 14](#_Toc87647315)

[**1.6.**  **Alternativas de Solución (Prototipos de Solución).** 14](#_Toc87647317)

[**1.7. Estado del arte** 16](#_Toc87647318)

[**CAPÍTULO 2 - MARCOTEÓRICO** 17](#_Toc87647319)

[**2.1.** **Marco teórico de programación orientada a objetos** 17](#_Toc87647320)

[**2.1.1** **NetBeans:** 17](#_Toc87647321)

[**2.1.2** **Java Swing** 17](#_Toc87647322)

[**2.1.3** **Java POO** 18](#_Toc87647327)

[**2.1.4** **IDEs** 18](#_Toc87647330)

[**2.1.5** **Entidad Relación: Diagrama de clases en Java** 19](#_Toc87647335)

[**2.1.6** **JavaScript** 22](#_Toc87647352)

[**2.1.7** **Variables** 23](#_Toc87647362)

[**2.1.8** **Mvc:** 23](#_Toc87647366)

[**2.1.9** **Framework Bootstrap:** 24](#_Toc87647371)

[**2.1.10** **SERVIDOR WEB** 25](#_Toc87647374)

[**2.1.11** **SERVIDOR DE BASE DE DATOS** 25](#_Toc87647376)

[**a.** **MYSQL** 25](#_Toc87647377)

[**2.1.12** **PLATAFORMA JEE (JAVA PLATFORM, ENTERPRISE EDITION)** 26](#_Toc87647378)

[**2.1.13** **HTML** 26](#_Toc87647379)

[**2.1.14** **CSS** 27](#_Toc87647380)

[**2.2.** **Marco teórico de las fórmulas y/o el sistema.** 27](#_Toc87647381)

**2.2.2 Sistema de cotización**…………………………………………………………………....27

**2.2.3 Pasos del proceso para calcular la función compra**……………………..…………27

[**CAPÍTULO 3 - DESARROLLO DE LA SOLUCIÓN** 27](#_Toc87647382)

[**3.1.** **Diagrama de clases** 27](#_Toc87647383)

[**3.2.** **Modelamiento de base de datos** 27](#_Toc87647384)

[**3.3.** **Entorno visual de los formularios** 27](#_Toc87647385)

[**3.4.** **Código Fuente** 27](#_Toc87647386)

[**CAPÍTULO 4 – RESULTADOS** 27](#_Toc87647387)

[**4.1.** **RESULTADOS DE LA ENCUESTA** 27](#_Toc87647388)

[**4.2.** **PRESUPUESTO** 28](#_Toc87647389)

[**Lienzo de propuesta de valor** 28](#_Toc87647390)

[**Project Charter** 28](#_Toc87647391)

[**MODELO CANVAS** 31](#_Toc87647392)

[**DIAGRAMA GANTT** 32](#_Toc87647393)

[**Bibliografias** 33](#_Toc87647394)

# **CAPÍTULO 1 ASPECTOS GENERALES**

# **Análisis de la empresa****:**

En Appsperu analizamos los desafíos del mercado y establecemos soluciones que permitan satisfacer las necesidades de nuestros clientes. Por ello, en este caso se desarrollará un sistema de carrito de compas en su página web para la juguetería Planet KIDS que se especializa en la venta de juguetes para niños con alguna discapacidad.

# **Visión**

Ser la juguetería más conocida por ser orientada a la comprensión de la diversidad de las personas. para las familias, niños y comunidades en la que ellos cumplan sus sueños y fantasías.

# **Misión**

A través de nuestra Juguetería, aspiramos a materializar las fantasías de nuestros clientes, brindando una sana entretención. Seremos una tienda especializada en ventas de juguetes para niños con alguna discapacidad para colaborar en la interacción social de las personas a través del intercambio de intereses, sueños y gustos.

# **Análisis del Entorno (PESTEL)**

El análisis PESTEL se utiliza para realizar un estudio del entorno que rodea a una empresa o sector. Consiste en un total de seis partes, cada una de las cuales hace referencia a un factor.

# **· Factor Político**

Una empresa dedicada a la tecnología tendrá que estar al tanto de si se mantiene la estabilidad política de los países en los que opera o si el gobierno altera leyes que afecten al negocio o los impuestos. La facilidad de convertir una idea de una aplicación en un negocio hace que este sea un factor a tener en cuenta. Por otro lado, hay que tener en cuenta que los gobiernos pueden prohibir o bloquear ciertas actividades, por ejemplo, Facebook está bloqueado en Irán, Vietnam y Corea del Norte y Twitter está prohibido en muchos otros países. Si las empresas utilizan estas redes sociales, deberán tener presente que no podrán llegar a cualquier parte del mundo, tendrán que considerar alternativas locales.

# **· Factor Económico**

Los impuestos, la tasa de desempleo, los indicadores de crecimiento económico o el cambio de divisas son elementos que se sitúan en esta categoría. Internet ha originado muchas opciones para determinar precios: subastas, precios dinámicos en función del stock, pagos mensuales en la venta de software... En las aplicaciones los modelos de negocio avanzan rápidamente y lo que antes era de pago, ahora es gratis con publicidad o con compras integradas.

# **· Factor Social**

Los aplicativos han cambiado la vida de los usuarios y cada vez incorporan más funcionalidades. En esta categoría habrá que tener en cuenta factores como los valores sociales, las tendencias demográficas, la movilidad en el trabajo, el nivel de educación y de adopción de la tecnología, la opinión de las personas con gran influencia, la aceptación de productos extranjeros, el impacto del idioma en la difusión de los productos, etc. Si se crea una aplicación con el objetivo de que esté disponible de manera global, habrá que tener en cuenta la cultura de otros países y adaptarla para que sea aceptada socialmente. Otras cuestiones a tener en cuenta son que no todo el mundo tiene acceso a internet y que países y grupos de edad diferentes tienen preferencias distintas en cuanto al uso de redes sociales o determinadas aplicaciones. Los desarrolladores de aplicaciones deberán conocer bien a su público objetivo y saber cómo llegar a él.

# **· Factor Tecnológico**

Los aplicativos se están imponiendo. Habrá que considerar las innovaciones tecnológicas relacionadas con la industria del software y la duración del ciclo de vida de la tecnología. Estar al tanto de los nuevos tamaños de pantalla será esencial para que la aplicación siempre se vea bien.

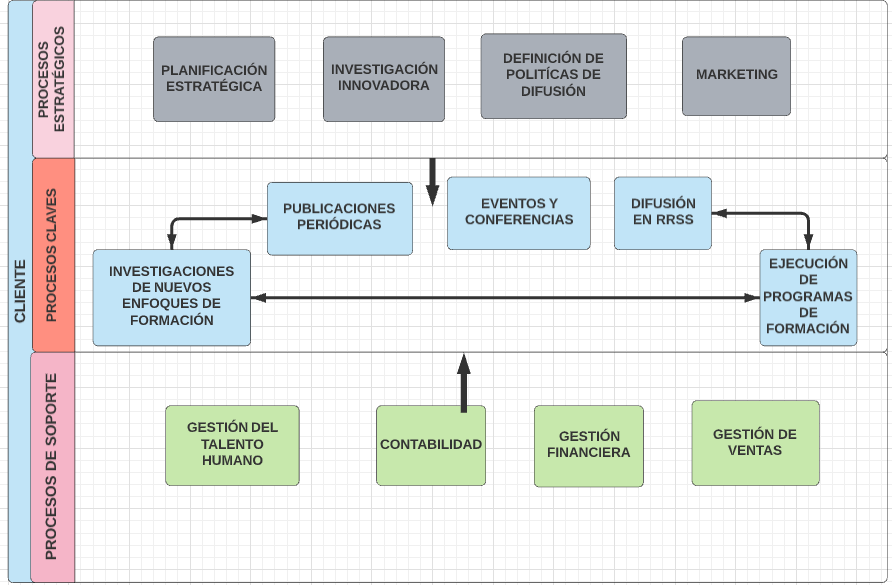
# **· Factor del Entorno**

Las empresas desarrolladoras de aplicaciones afectan al medio ambiente debido al gasto energético. Además, es recomendable utilizar sistemas de gestión documental que permitan reducir el uso del papel.

# **· Factor Legal**

Habrá que estar al tanto de las leyes que puedan afectar a la empresa, por ejemplo, las de emprendimiento, si fuera el caso, pero también de las normas específicas de la industria del software. También habrá que saber cómo proteger la aplicación legalmente: patentes y derechos de autor. Adicionalmente, hay varias leyes que también pueden afectar a la empresa: violación de copyright, utilización de nombres de marcas populares, pago secreto a creadores de opinión, venta de datos de los clientes, etc.

# **Mapa de procesos**



# **Estrategia de Negocio (Tablero de Control de Mando)**

# **Modelo Canvas**

# **Lienzo de propuesta de valor**

Diagrama

Descripción generada automáticamente

# **1.2. Descripción del Problema**

En la Actualidad, el sector de jugueterías uno de los servicios con demanda en el mercado. Existen miles de niños en la sociedad que sienten una gran necesidad de adquirir un juguete entretenido y padres dispuestos a gastar su dinero con tal de ver sus hijos felices, lo que representa una excelente ventaja a la hora de captar a los niños como clientes potenciales.

El problema es el desconocimiento de donde compra un juguete especializado con productos para niños con discapacidad . Para ello, se desarrollará este carrito de compras que es un aplicación web que facilita a los consumidores la compra de algún producto al aceptar el pedido y el pago correspondiente en pocos pasos, con diferentes formas de cobro y reserva del producto.

**1.3.** **Definición de Objetivos**

## **1.3.1** **Objetivo General**

## Se diseñara un aplicación web de carrito de compras que facilite a los consumidores la compra de algún producto que acepte el pedido y el pago correspondiente en pocos pasos, con diferentes formas de pago.

## **1.3.2** **Objetivos Específicos**

* Diseñar Diagrama de clases.
* Realizar Mapa de Procesos
* Realizar Estrategia de Negocio
* Diseñar el Modelo de la Base de datos
* Diseñar el Modelo Canvas
* Diseñar el Diagrama de Gantt.
* Realizar un diseño de una interfaz información clara y fácil de entender
* Realizar la aplicación con Java y el método MVC
* Realizar el módulo de carrito de compras.

**1.4 alcances y limitaciones**

**1.4.1 Alcances**

* Documento Diagrama de clases.
* Documento Mapa de Procesos
* Documento Estrategia de Negocio.
* Documento del Modelo de la Base de datos.
* Documento del Modelo Canvas.
* Documento del Diagrama de Gantt.

**1.4.2 Limitaciones**

* El aplicativo web se programara en java, JavaScript. con complementos cómo HTML, Css y Bootstrap.
* El aplicativo web se realizara bajo un cronograma de proyecto del documento del diagrama de Gantt

**Requisitos Funcionales:**

* El aplicativo web debe permitir iniciar sesión al cliente y crear una nueva cuenta.
* El aplicativo web debe permitir la selección del producto a comprar y agregar al carrito de compra.
* El aplicativo web debe permitir la modificación del producto en el carrito de compra como la eliminación, editar la cantidad del producto.
* El aplicativo web en el módulo de carrito de compra realizará el cálculo correspondiente de los productos comprados y podrá realizar la compra.

# **1.5.** **Justificación**

## Este proyecto tiene como principal propósito atenuar los problemas que los padres encuentran al momento de conseguir un juguete para su niño mediante la creación de un aplicativo web que permita el juguete adecuado de forma inmediata, todo esto de gran importancia, puesto que los padres no tendrán mayor problema en elegir uno de los juguetes.

# **1.6. Alternativas de Solución (Prototipos de Solución).**

**Primera propuesta**

Captura de pantalla de computadora

Descripción generada automáticamente

Una captura de pantalla de una computadora

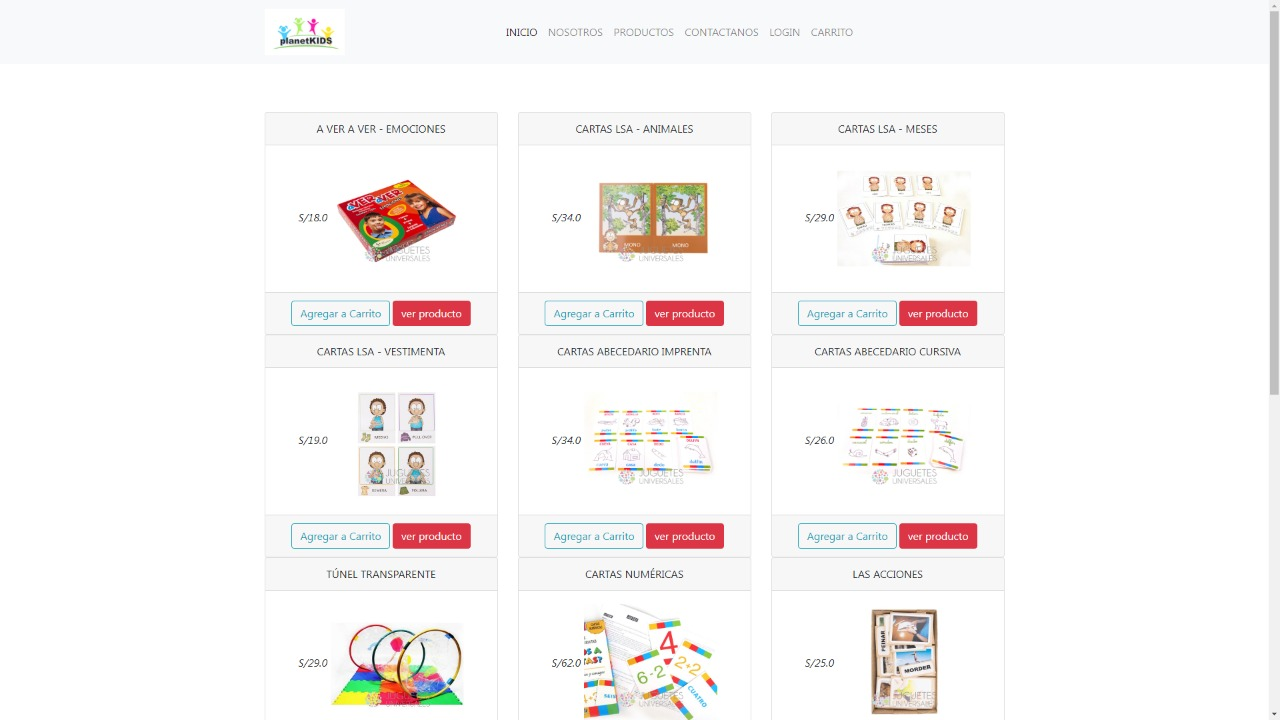
Descripción generada automáticamente

**Segunda propuesta**

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

**Tercera propuesta**



# **1.7. Estado del arte**

* Según Vilaplana, F. 2017 en su artículo con respecto a la digitalización determina que esta debe estar liderada por el aprendizaje continuo y donde las personas puedan desarrollar sus aptitudes y encontrar el entorno adecuado para fomentar su creatividad y donde la tecnología sirva de palanca para conseguir los objetivos propuestos. Para que se cumplan las exigencias anteriormente mencionadas hay que acompañarlo de una infraestructura tecnológica. El aporte de Vilaplana nos muestra la importancia de la tecnología para mejorar el rendimiento, pero con la observación de que esta tecnología es un soporte para las habilidades humanas que se deben cultivar en las personas.
* Los estudios realizados por Vita, N. 2008 en su artículo sobre la importancia de las tecnologías de la información de las TICs considera que las tecnologías de información y comunicación contribuyen en la recepción de la información en las mejores condiciones técnicas y en el menor tiempo posible.

Los computadores no solo sirven para procesar información sino también como medio de comunicación interactiva entre la empresa y el cliente. Este cambio se está produciendo a gran escala en todos los ámbitos socioeconómicos y de las actividades humanas. Por eso, nuestra organización no se puede quedar atrás y debe tomar decisiones firmes para implementar la tecnología a sus procesos administrativos, operativos y comerciales.

* Para Olortegui Melendez, Luis 2016 nos da a entender en tesis “sistema de información web para mejorar la gestión comercial de la empresa librería lizdaroni de pacanguilla” demostrar que se puede mejorar la gestión comercial de una empresa con la implementación de un sistema que permitirá a dicha empresa automatizar los procesos de control de compras, ventas y Almacén, logrando la reducción de tiempos y costos de procesamiento. También nos dice en su tesis que podemos tener una visión clara y objetiva de lo que realmente necesita esta organización, tomando como punto de referencia las opiniones tanto del personal como del gerente de dicha empresa, y la observación realizada en las visitas hechas a la misma.

# **CAPÍTULO 2 - MARCOTEÓRICO**

# **Marco teórico de programación orientada a objetos**

1. **NetBeans**

Es una herramienta para el uso de aplicaciones Web, móvil y de Escritorio, asimismo utiliza diferentes lenguajes de programación como Java, C++, Ruby y PHP entre otros. Se considera una plataforma de código abierto, también multiplataforma, multilenguaje, contiene servidores web y el entorno es amigable para el usuario. (Mendoza, 2008)

1. **Java POO**

La programación orientada a objetos establece un equilibrio entre la importancia de los procesos y los datos, mostrando un enfoque más cercano al pensamiento del ser humano. Se introduce un aspecto novedoso respecto al anterior paradigma: la herencia, facilitando el crecimiento y la mantenibilidad.

Las bases de la programación orientada a objetos son: abstracción, encapsulación, modularidad y jerarquización. (Bermúdez, 2012)

1. **IDEs**

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDES pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes

Los IDES ofrecen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como: C++, Python, Java, C#, Delphi, Visual Basic, etc.

.

**LOS 10 MEJORES IDES**

**Imagen que contiene objeto, botiquín de primeros auxilios, reloj

Descripción generada automáticamente**

1. **Entidad Relación: Diagrama de clases en Java**

**Diagrama de clases**

El diagrama de clases es uno de los diagramas incluidos en UML clasificado dentro de los diagramas de estructura y, como tal, se utiliza para representar los elementos que componen un sistema de información desde un punto de vista estático.

Es importante destacar que, por esta misma razón, este diagrama no incluye la forma en la que se comportan a lo largo de la ejecución los distintos elementos, esa función puede ser representada a través de un diagrama de comportamiento, como por ejemplo un diagrama de secuencia o un diagrama de casos de uso.

El diagrama de clases es un diagrama puramente orientado al modelo de programación orientado a objetos, ya que define las clases que se utilizarán cuando se pase a la fase de construcción y la manera en que se relacionan las mismas. Se podría equiparar, salvando las distancias, al famoso diagrama de modelo Entidad-Relación (E/R), no recogido en UML, tiene una utilidad similar: la representación de datos y su interacción. Ambos diagramas muestran el modelo lógico de los datos de un sistema. (Francisco José García Peñalvo)

**Relaciones de agregación:** las relaciones de agregación se basan en la idea de observar o entender un objeto como una composición de otros objetos, desde nuestro punto de vista, las relaciones de agregación se entenderán como relaciones en las cuáles una serie de clases aparecen como tipos de los atributos de otra clase,

Estas relaciones se conocen también como relaciones “todo - partes”. El “todo” está representado por la clase que aglutina a las otras clases, y las “partes” están dadas de diversas clases que aparecen.

La mejor forma de identificar si nos encontramos ante una relación de agregación es preguntarnos si la clase que queremos definir “tiene un” (en inglés, “has - a”) atributo de la otra clase que estemos usando (de ahí que en ciertas referencias se definen como relaciones “has - a”). Fernando Berzal

Diagrama

Descripción generada automáticamente

Diagrama

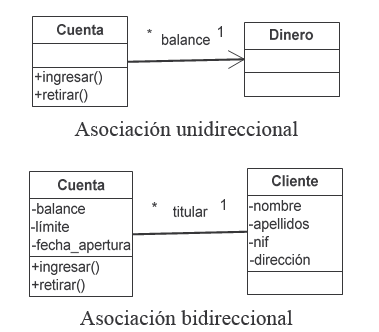
Descripción generada automáticamente

**Relaciones de Asociación:** cuando dos o más clases tiene relación de asociación cuando una de ellas tenga que requerir o utilizar alguno de los servicios (es decir, acceder a alguna de las propiedades o métodos) de las otras.

Las relaciones de asociación se crean enlaces entre objetos. Estos enlaces no tienen por qué ser permanentes (es más, en la mayoría de los casos, no lo serán). Los objetos deben tener entidad fuera de la relación (a diferencia de las relaciones de composición).

(Fernando Berzal)

Ejemplo:



**Multiplicidad de las asociaciones**

La multiplicidad de una asociación determina cuántos objetos de cada tipo intervienen en la relación:

* Cada asociación tiene dos multiplicidades (una para cada extremo de la relación).
* Para especificar la multiplicidad de una asociación hay que indicar la multiplicidad mínima y la multiplicidad máxima (mínima. Máxima)
* Cuando la multiplicidad mínima es 0, la relación es opcional.
* Una multiplicidad mínima mayor o igual que 1 establece una relación obligatoria. (Fernando Berzal)

Tabla

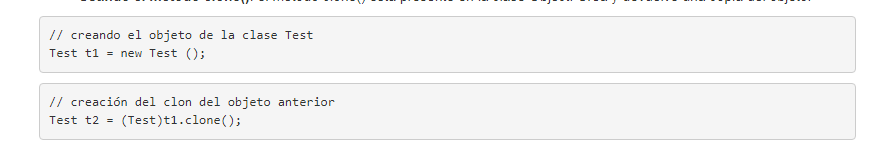
Descripción generada automáticamente

1. **Java**

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para JavaScript. Si bien su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet). Java es un lenguaje.

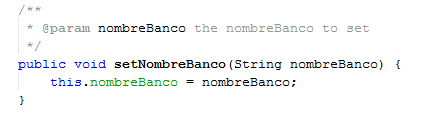
**Objeto**

Se le denomina objeto a toda la plantilla que contenga atributos y métodos para ser utilizadas al momento en que se crean.



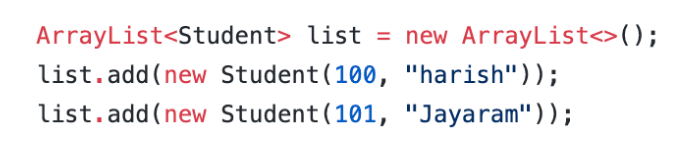
**Encapsulamiento:**

Es el proceso que consiste en organizar los datos y operaciones (métodos) de una clase que constituyen su estructura y su comportamiento con el fin de evitar el acceso a datos por cualquier otro medio que no sea el especificado y por esto el encapsulamiento de datos asegura la integridad de los datos que contiene el objeto



**Arraylist**

La clase ArrayList en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.



**Variables**

Es una variable de instancia o miembro de dato es una variable que se relaciona con una única instancia de una clase.

Cada vez que se crea un objeto, el sistema crea una copia de todas las variables que están vinculadas con dicha clase, haciéndolas propias de esa instancia. Solamente se puede acceder a ellas a través del identificador del objeto.

Estas variables son declaradas fuera del cuerpo de los métodos y dentro de la clase, por lo tanto, son de tipo global. Por ende, pueden ser utilizadas por cualquier método no estático de dicha clase.

**Mvc:**

Es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. ​ Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

* **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tantas consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).
* **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
* **Vista**: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto, requiere de dicho 'modelo' la información que debe representar como salida.

1. **Framework Bootstrap:**

Para el desarrollo de esta aplicación web utilizamos Bootstrap. Este Framework Front-end es gratuito, es fácil de usar y muy flexible, pues puede adaptarse a casi todas las necesidades de desarrollo web Front-end. Sus mejores capacidades incluyen, entre otras, brindar funciones de respuesta que permite que la interfaz de usuario de un sitio web funcione de manera óptima en todos los tamaños de pantalla, ya sea en teléfonos de pantalla pequeña o en dispositivos de escritorio de pantalla grande.

Bootstrap contiene todo tipo de plantillas de diseño basadas en HTML y CSS para diversas funciones y componentes, como navegación, sistema de cuadrícula, carruseles de imágenes y botones.

1. **SERVIDOR WEB**

Un servidor web (server) es un ordenador de gran potencia que se encarga de “prestar el servicio” de transmitir la información pedida por sus clientes. Los servidores web (web server) son un componente de los servidores que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP (Hipertext Transfer Protocol).

1. **SERVIDOR DE BASE DE DATOS**

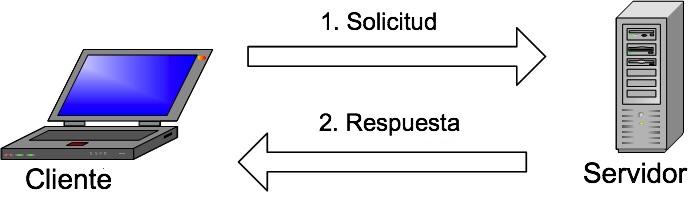
Un servidor de base de datos, también conocido como database server o RDBMS (Relational DataBase Management Systems) en caso de bases de datos relacionales, es un tipo de software de servidor que permiten la organización de la información mediante el uso de tablas, índices y registros.

1. **MYSQL**

Es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional.

Base de datos es una colección de datos estructurados. Una base de datos es un lugar en el que los datos son almacenados y organizados. La palabra «relacional» significa que los datos almacenados en el conjunto de datos son organizados en forma de tablas. Cada tabla se relaciona de alguna manera. Si el software no es compatible con el modelo de datos relacionales, simplemente se llama DBMS.

Explica la estructura básica cliente-servidor. Uno o más dispositivos (clientes) se conectan a un servidor a través de una red específica. Cada cliente puede realizar una solicitud desde la interfaz gráfica de usuario (GUI) en sus pantallas, y el servidor producirá el output deseado, siempre que ambas partes entiendan la instrucción.

****

1. **PLATAFORMA JEE (JAVA PLATFORM, ENTERPRISE EDITION)**

Es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE.

JEE simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchas de las funciones de la aplicación

de forma automática, sin necesidad de una programación compleja.

1. **HTML**

Corresponde a las siglas de HyperText Markup Language (lenguaje de marcado de hipertexto), es el lenguaje predominante en la elaboración de páginas Web que nos permite tener documentos estructurados y vinculados entre sí. Es usado para describir la estructura y el contenido en forma de texto, así como para completar el texto con objetos tales como imágenes, tablas.

1. **JAVASCRIPT**

Es un leguaje se secuencias de comandos que te permite crear contenido de actualización dinámica controla multimedia animar imágenes y practicante todo lo demás.

Las funciones son uno de los bloques de construcción fundamentales en JavaScript. Una función JavaScript es similar a un procedimiento un conjunto de instrucciones que realiza una tarea o calcular un valor, pero para que un procedimiento califique como función, debe entre la entrada y la salida. Para usar una función, debes definir la en algún ligar en el ámbito desde el que deseas llamarla.

1. **CSS**

Las hojas de estilo en cascada o CSS, Cascading Style Sheets, es un lenguaje utilizado para definir la presentación de un documento estructurado, HTML o XHTML.

Este tipo de lenguaje nos permite de una forma sencilla, definir todo tipo de formatos, colores… en definitiva la apariencia que le queremos dar a nuestra aplicación Web, inclusive podemos disponer de varias hojas de estilo, para que según que visitante o que usuario, según el navegador con el que accedemos, si queremos imprimir una página… Los cambios se realizan en una única página, no tenemos que ir página por página definiendo un nuevo estilo, con cambiar nuestro CSS es suficiente, es una forma muy efectiva y potente.

1. **MAVEN**

Maven es una herramienta open-source, que se creó en 2001 con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente).

Antes de que Maven proporcionara una interfaz común para hacer builds del software, cada proyecto solía tener a alguna persona dedicada exclusivamente a configurar el proceso de build.

Si queríamos compilar y generar ejecutables de un proyecto, teníamos que analizar qué partes de código se debían compilar, qué librerías utilizaba el código, dónde incluirlas, qué dependencias de compilación había en el proyecto.

Ahora, la build de cualquier proyecto Maven, independientemente de sus módulos, dependencias, librerías…consiste simplemente en ejecutar el comando mvn install.

Como vemos, Maven simplifica mucho el proceso de build del código, permitiéndonos compilar cualquier tipo de proyecto de la misma manera, librándonos de todas las dificultades que hay por detrás.

Pero lo cierto es que Maven es mucho más que una herramienta que hace builds del código.

Podríamos decir, que Maven es una herramienta capaz de gestionar un proyecto software completo, desde la etapa en la que se comprueba que el código es correcto, hasta que se despliega la aplicación, pasando por la ejecución de pruebas y generación de informes y documentación.

Para ello, en Maven se definen ciclos de build del software con una serie de etapas diferenciadas.

Por ejemplo, el ciclo por defecto tiene las etapas de:

* **Validación (validate):** Validar que el proyecto es correcto.
* **Compilación (compile).** Genera los ficheros .class compilando las fuentes .java
* **Test (test):** Probar el código fuente usando un framework de pruebas unitarias.
* **Empaquetar(package):** Empaquetar el código compilado realizado y transformarlo en algún formato tipo .jar o .war.
* **Pruebas de integración (integration-test):** Procesar y desplegar el código en algún entorno donde se puedan ejecutar las pruebas de integración.
* **Verificar(verify):** que el código empaquetado es válido y cumple los criterios de calidad.
* **Instalar(install):** el código empaquetado en el repositorio local de Maven, para usarlo como dependencia de otros proyectos.
* **Desplegar(deploy):** Copia el ficher.jar a un servidor remoto, poniéndolo disponible para cualquier proyecto Maven con acceso a ese servidor remoto.

Por otra parte, con Maven la gestión de dependencias entre módulos y distintas versiones de librerías se hace muy sencilla. En este caso, solo tenemos que indicar los módulos que componen el proyecto, o qué librerías utiliza el software que estamos desarrollando en un fichero de configuración de Maven del proyecto llamado POM.

Además, en el caso de las librerías, no tienes ni tan siquiera que descargarlas a mano. Maven posee un repositorio remoto (Maven central) donde se encuentran la mayoría de librerías que se utilizan en los desarrollos de software, y que la propia herramienta se descarga cuando sea necesario.

Este es un ejemplo de POM donde puedes ver todo esto:

Como puedes ver, en el mismo archivo también indicamos aspectos como el nombre del proyecto, licencias, quién desarrolla el código, el sitio web, el repositorio de control de versiones, un identificador único del proyecto etc.

Digamos que Maven aporta una semántica común al proceso de build y desarrollo del software.

1. **JASPER REPORTS**

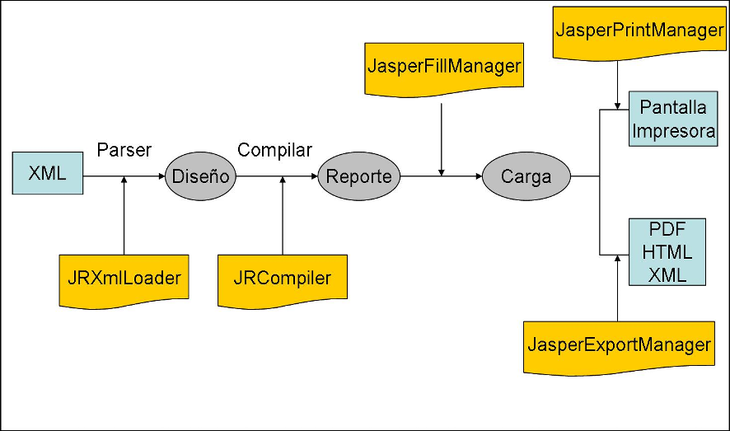
JasperReports es un software diseñado de modo que está compuesto por un conjunto de librerías java que tienen la función de facilitar la generación de informes en nuestras aplicaciones tanto Web como de escritorio.

JasperReports es el sistema de motor de informes de código abierto más popular del mundo. Está escrito completamente en Java y es capaz de utilizar los datos procedentes de cualquier tipo de fuente de datos y presentar los documentos con precisión de píxel, dichos documentos se pueden ver, imprimir o exportar en una variedad de formatos de documentos.

Los informes se definen en un fichero XML el cual será compilado por las librerías Jasper Report para generar un fichero. jasper que utilizaremos para rellenar y mostrar el informe final. La salida de los informes pueden ser PDF, CVS, XML, TXT, HTML, XLS, RTF, Jasper Viewer, y muchos otros.

Características y funcionalidades principales del software JasperReports:

* Paneles, mesas, tablas de referencias cruzadas, gráficos y medidores
* informes basados ​​en Web y píxel perfecto
* Informe de salida en PDF, XML, HTML, CSV, XLS, RTF, TXT
* Soporte integrado de códigos de barras
* Visual rotación de texto
* Estilos de la colección
* No hay límite al tamaño de informe
* Impresión condicional

presentamos un modelo gráfico con la secuencia

# **Marco teórico de las fórmulas y/o el sistema.**

EL Software de sistema de ventas está orientado a la Gestión de Empresas Comerciales, es un Software Práctico, Exacto y Seguro que le permitirá tener un mejor control a nuestra Empresa. Este se trata de una completa aplicación, para la gestión de clientes y productos, donde se incluye la posibilidad que el cliente pueda elegir el método de pago que desea recibir como constancia de su compra, la aplicación también puede realizar el registro de ventas de dichos productos.

Es una secuencia de pasos que emprende el vendedor para tratar con un comprador potencial y que tiene por objeto producir alguna reacción deseada en el cliente. En este sistema de ventas del producto, se evalúan las características y las necesidades de promoción del servicio y posteriormente se contacta con los clientes para cerrar las ventas, tomando pedidos y llevando los productos seleccionados de manera que satisfacen la necesidad de nuestro cliente.

De este modo el departamento de ventas del negocio mantiene la base de datos actualizada sobre las ventas realizadas, los pedidos en proceso y puede comunicarse con el administrador de la tienda y pedidos en almacén para poder ser atendidos a tiempo y de manera eficaz, se puede encontrar estos dos tipos de comercialización que son los siguientes:

Tienda online: es un tipo de comercio electrónico que consiste en que la empresa desarrolle su propia página web para la venta de sus productos, o también en que, dentro de su página corporativa, desarrolle una sección especializada para la tienda digital.

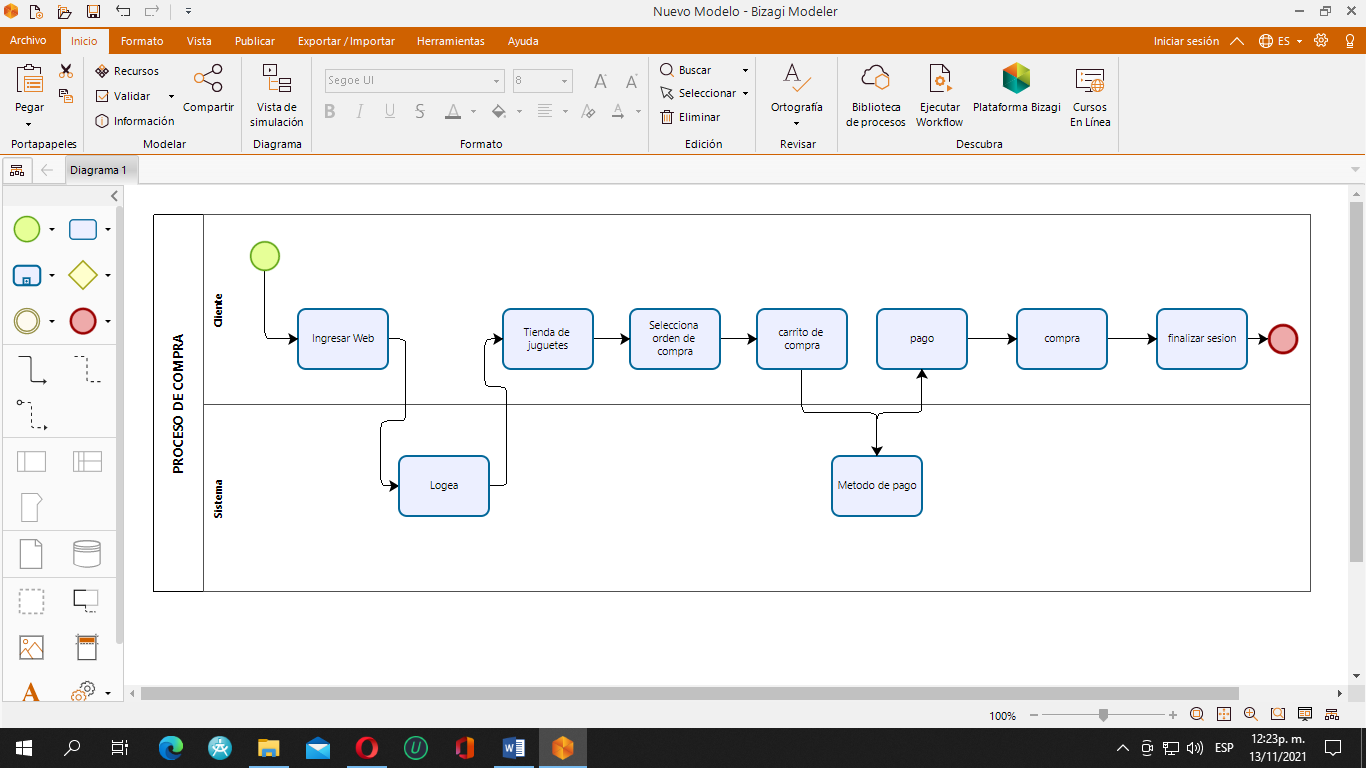
Marketplace: este tipo de comercio electrónico hace referencia a aquellas plataformas en las que cualquier usuario u empresa puede comprar y vender productos. En este sentido, los Marketplace se diferencian de las tiendas online en el hecho de que la oferta de mercancías no es realizada por un único vendedor o [negocio electrónico](https://www.beetrack.com/es/blog/negocios-electronicos-transporte).

**2.2.2 Sistema de cotización**

Consiste en una serie de procesos para poder realizar la orden y compra de juguetes para niños con discapacidad, este sistema permite interactuar con el cliente para dar una mejor funcionalidad y satisfacción al momento de realizar una compra

**2.2.3 Pasos para el proceso para Sistema de compra**

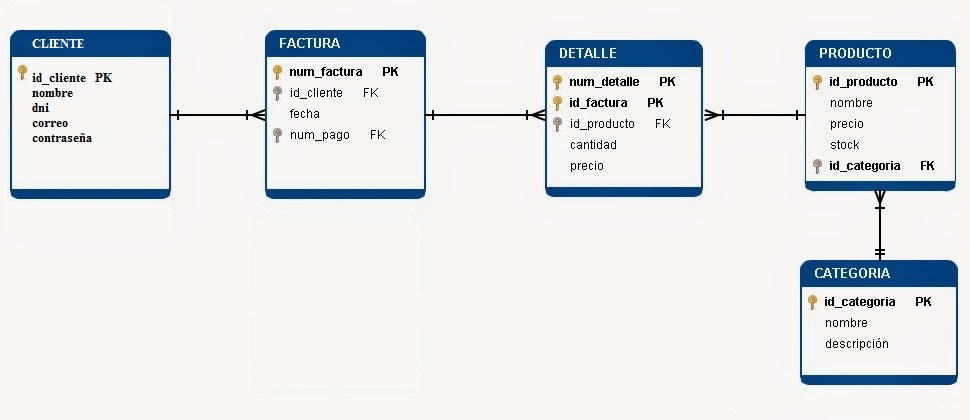
* paso 1: El usuario ingresa a la página web.
* paso 2: El usuario se inicia sesión e ingresa a la web de usuario
* paso 3: el usuario ingresa a la pestaña de productos donde se encuentra la sección de tienda web.
* paso 4: el usuario escoge los juguetes que va ordenar y se agrega al carrito de compras.
* paso 5: el usuario va a la pestaña carrito de compras y verifica la los productos agregados.
* paso 6: el usuario hace el realiza compra con el método de pago que desea el usuario.
* paso 7: el usuario verifica la da a comprar el producto y regresa a la pestaña principal.
* paso 8: en caso quiera seguir comprando regresa al paso 3, si no, se cierra sesión y sale de su cuenta.

**Diagrama de flujo**

# **CAPÍTULO 3 - DESARROLLO DE LA SOLUCIÓN**

# **Diagrama de clases**

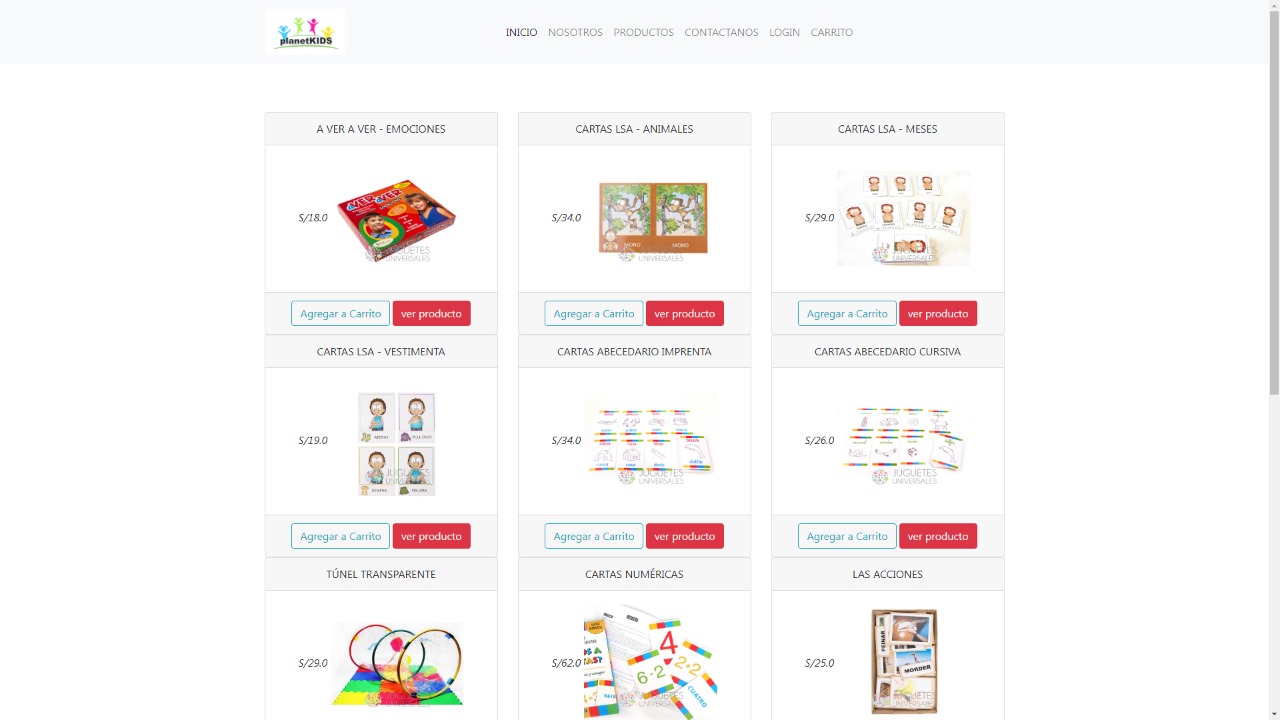
# **Modelamiento de base de datos**

****

# **Entorno visual de los formularios**

**HOME**

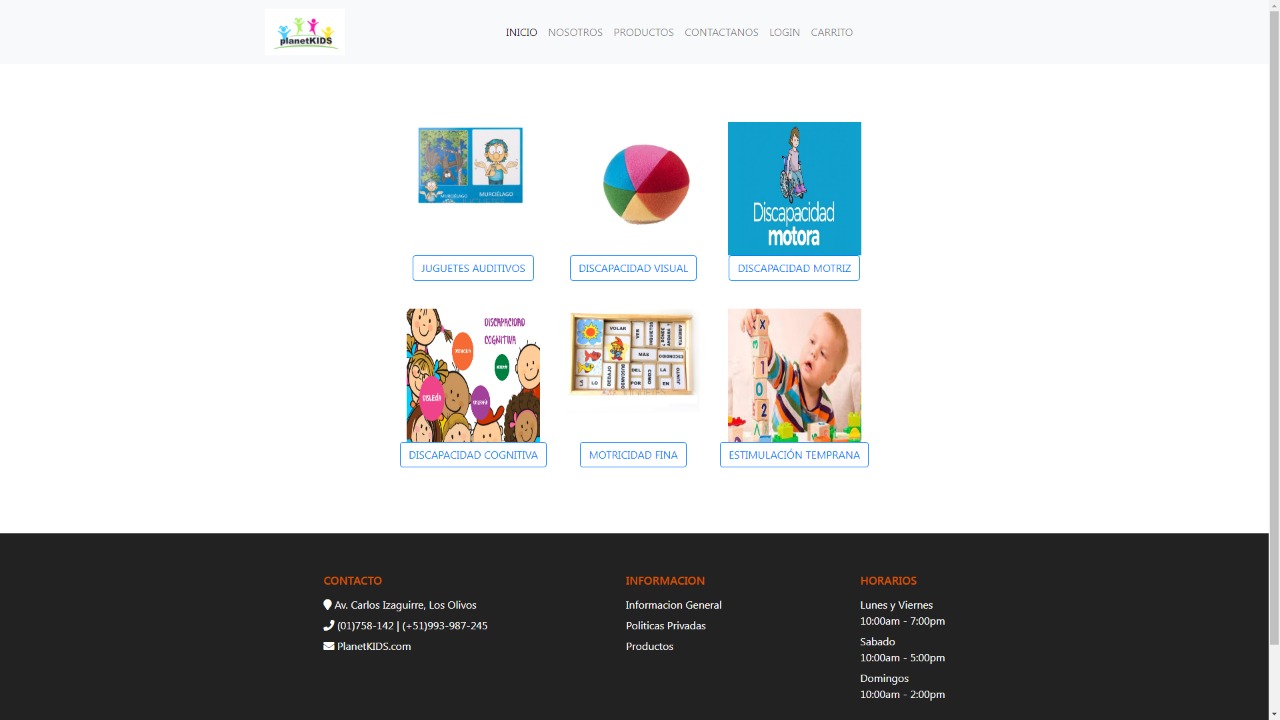
**PRODUCTOS**

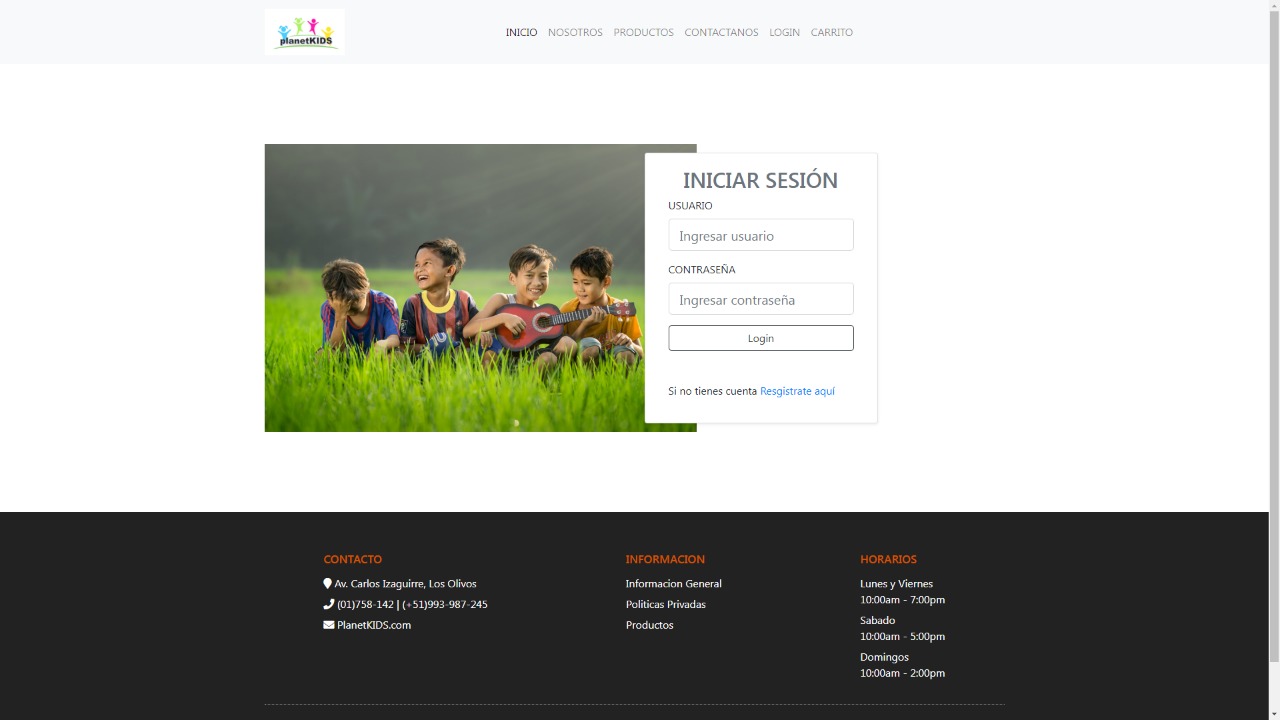


**CARRITO DE COMPRA**

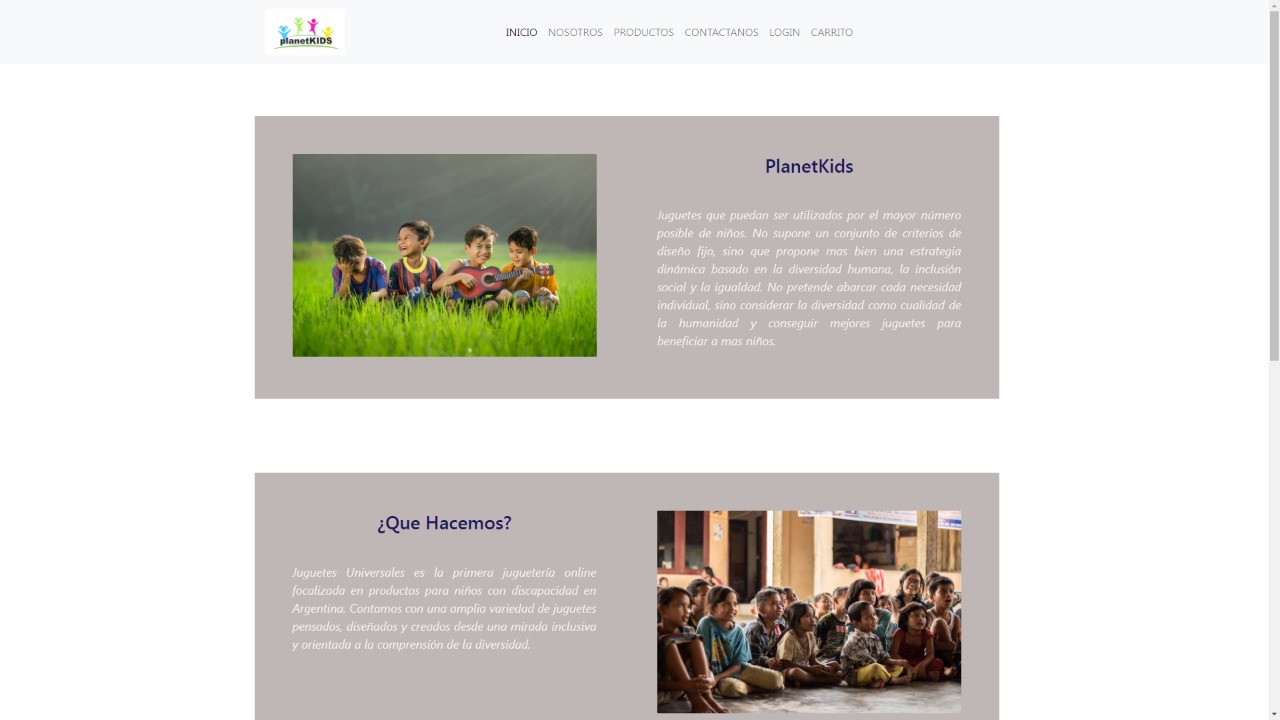
**REGISTRO**

**CATEGORIA**

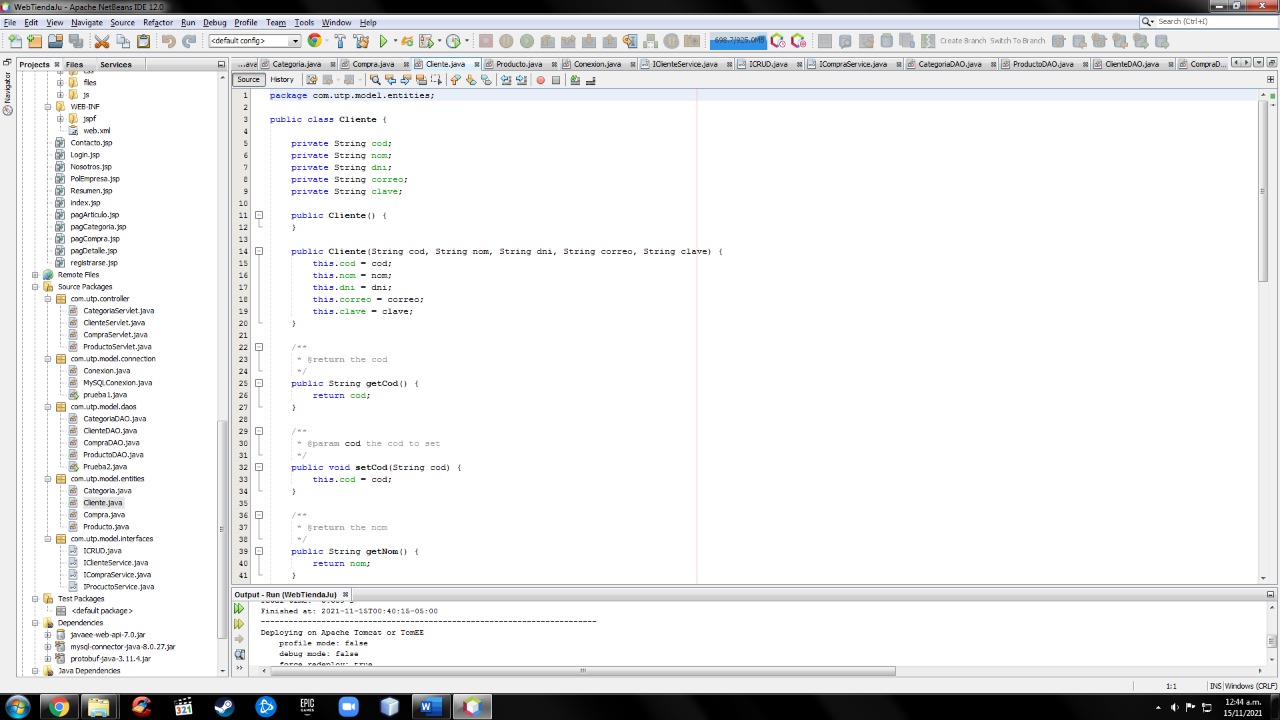


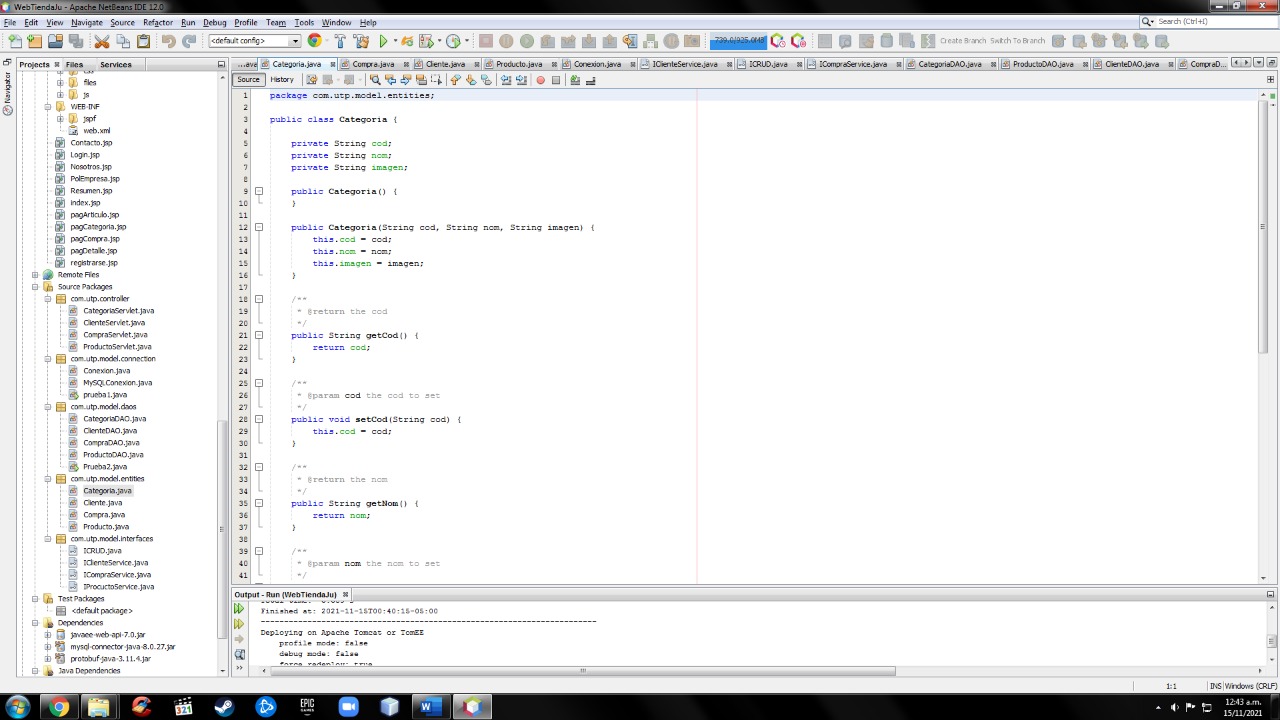
**LOGIN**

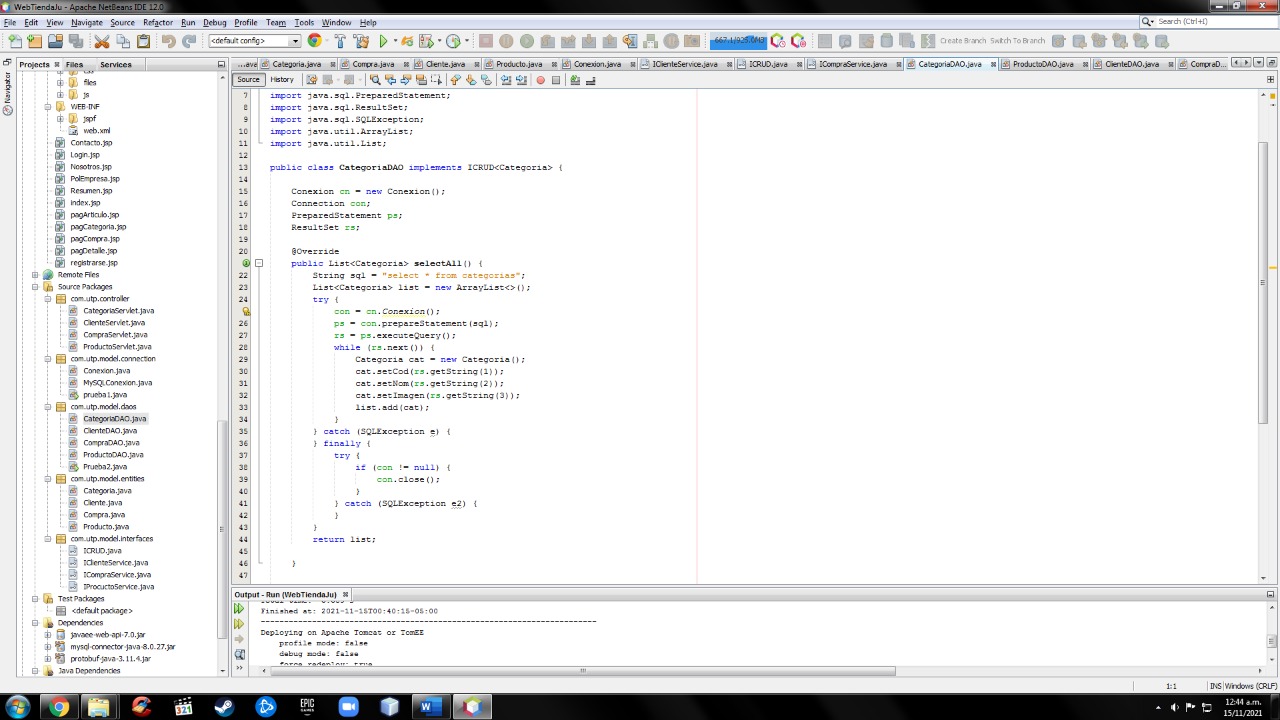
**NOSOTROS**

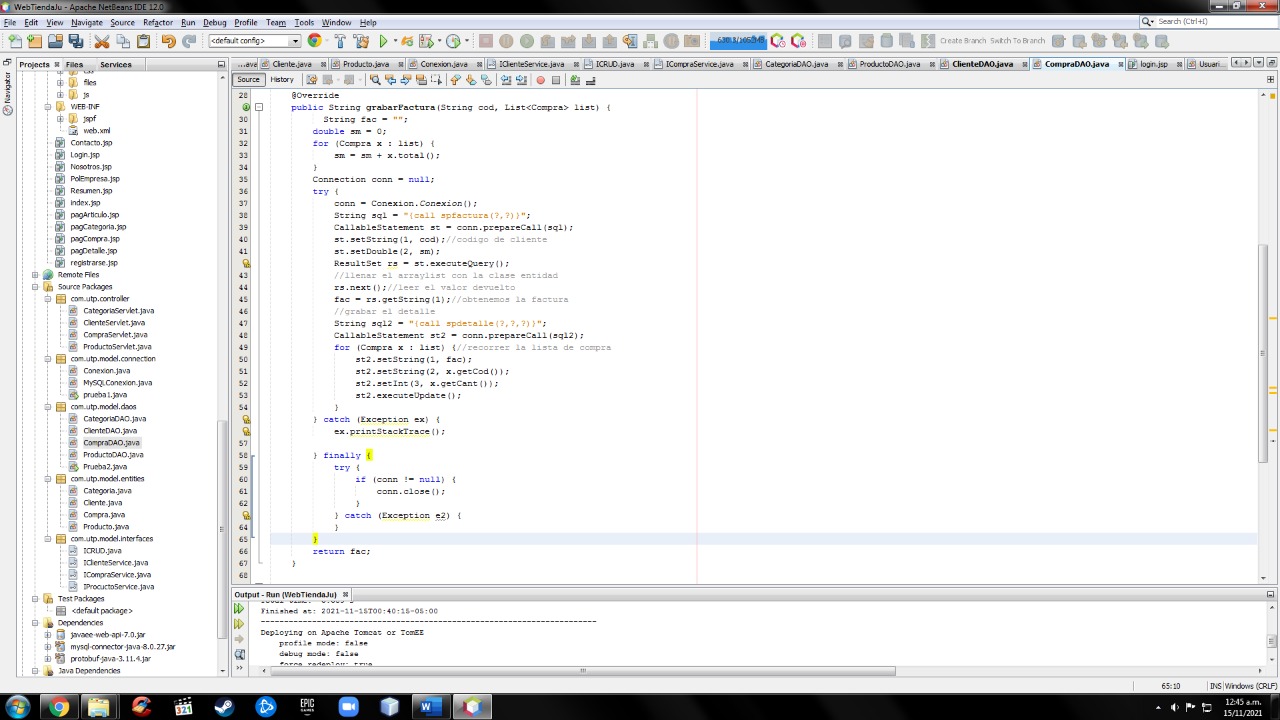


# **Código**

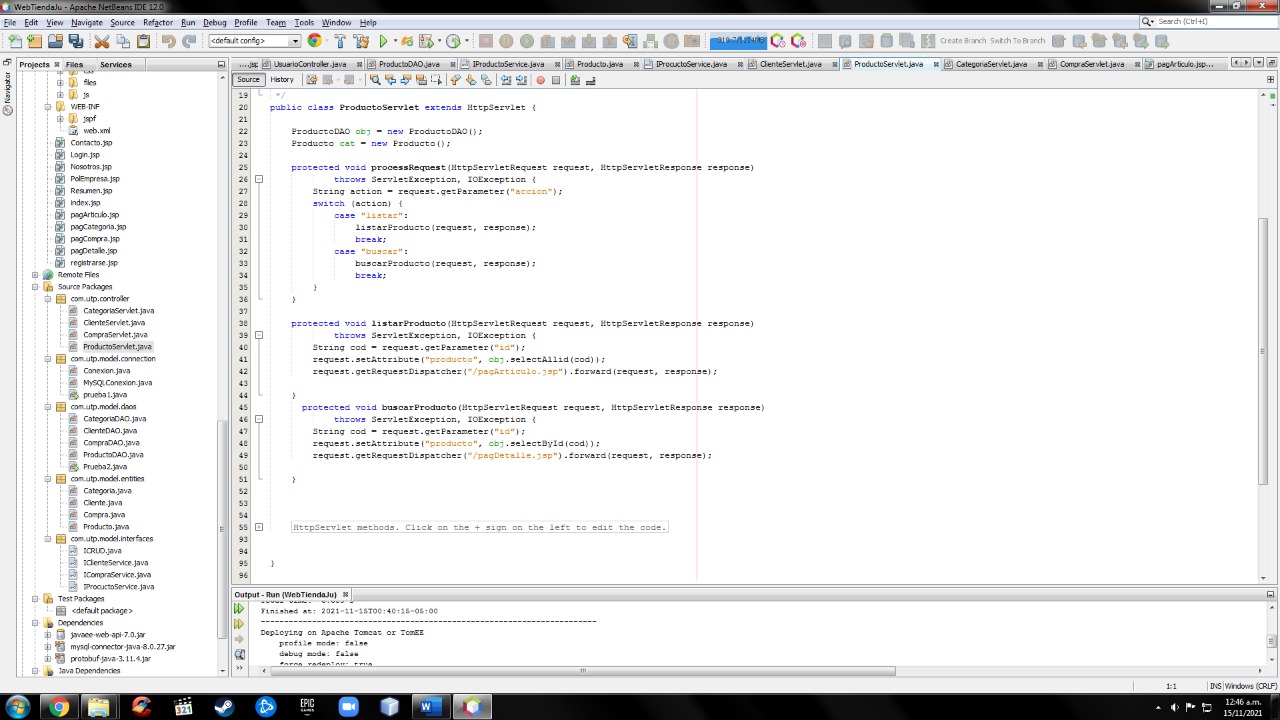


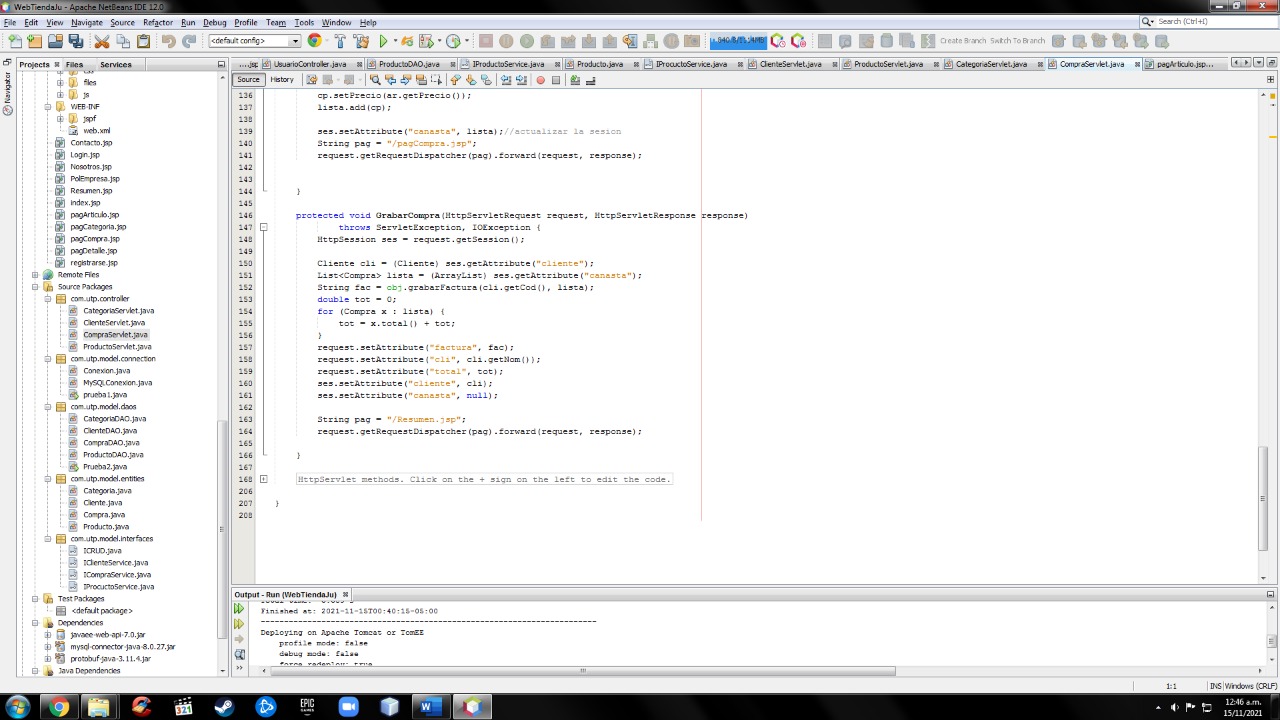












# 

# **CAPÍTULO 4 – RESULTADOS**

# **RESULTADOS DE LA ENCUESTA**

# **PRESUPUESTO**

anexo

# **Lienzo de propuesta de valor**

Diagrama

Descripción generada automáticamente

# **Project Charter**

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

# **MODELO CANVAS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SOCIOS CLAVE**   * Socios tecnológicos | **ACTIVIDADES CLAVE**   * Marketing digital * Desarrollo y mantenimiento   de la página web. | **PROPUESTA DE VALOR**   * Juguetes con una mirada inclusiva orientada a la compresión de la diversidad. * Juguetes creados desde una mirada inclusiva y orientada a la diversidad | | **RELACIONES CON CLIENTES**   * Páginas webs. * Redes sociales. * Foros. * Correos. | **SEGMENTOS DE CLIENTES**   * Nos focalizamos en personas que buscan productos especializados para niños con alguna discapacidad. |
| **RECURSOS CLAVE**  **Recursos Humanos**   * Programadores back-end. * Programadores front-end. * Especialista en SEO.   **Recursos Físicos**   * Espacio de trabajo. * Computadoras, Laptops. | **CANALES**   * Página web * Aplicativo web |
| **ESTRUCTURA DE COSTOS**   |  |  | | --- | --- | | **Costos fijos:**   * Pagos de servicios básicos. * Alojamiento web. * Nombre del dominio. | **Costos variables:**   * Pago personal operativo. | | | | **FLUJOS DE INGRESOS**   * Publicidad | | |

***Título del Proyecto: Pagina web juguetería Planet KIDS***

# **DIAGRAMA GANTT**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

# **Bibliografias**

<https://repositorio.uncp.edu.pe/handle/20.500.12894/1153?show=full>

<https://dspace.unitru.edu.pe/handle/UNITRU/5277>

Naya De Vita Montiel. (2008). TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN PARA LAS ORGANIZACIONES DEL SIGLO XXI. En Vol. 5 Núm. 1(77-86). Maracaibo: Instituto Universitario de Tecnología de Maracaibo.

Francisco Vilaplana, Guido Stein. (2020). Digitalización y personas. En REVISTA EMPRESA Y HUMANISMO(pp.9-12). Madrid: IESE Madrid.