



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
CALIDAD DE SOFTWARE

**JUEGO DE EMPAREJAMIENTO PARA PROYECTO DE
ORIENTACIÓN ESPACIAL v1.0**

INTEGRANTES: EMILIO ARMAS

WILLIAM ENRIQUEZ

JUAN PROAÑO

MICHAEL TROCELLIER

JIMMY VALLADARES

GRUPO: ZenQuality Labs

DOCENTE: SANDRA SANCHEZ, PhD

CURSO: GR2SW

FECHA: 8/11/2024

PERIODO

2024-B

Indice

Indice	2
Historias de usuario	4
Primera iteración.....	4
Segunda iteración:	6
Tercera iteración:	8
Herramientas.....	10
Visual Studio Code (VS Code).....	10
Git y GitHub	10
Mermaid.js	11
Tecnologías	11
Node.js	11
Express.js	11
Sequelize.....	11
MySQL	11
HTML,	11
Arquitectura	12
Diagrama de Arquitectura MVC	12
Diseño	13
Diagrama Entidad-Relación (ER).....	13
Diagrama de Clases	14
Especificación de casos de prueba	15
Caso de Prueba: CP01	15

Caso de Prueba: CP02	17
Caso de Prueba: CP03	18
Caso de Prueba: CP04	20
Caso de Prueba: CP05	23

Historias de usuario

Primera iteración

Esta iteración busca dar la pauta inicial del producto que se busca crear. Se enfoca en el uso del doctor ya que este será el profesional capacitado para que aplique la actividad con el público a quién se encuentra destinado el proyecto.

Nro: HU-001	Título: Visualización y asociación de productos en pantalla de juego	Prioridad: Alta Estimacion: 10 horas
Historia de usuario: Como Doctor quiero que haya 16 productos para relacionarlas con un sitio y profesional para que el paciente pueda Jugar		
Criterios de aceptación: 1) Dado que un paciente esté en la pantalla del juego Cuando el paciente vea la lista de productos Entonces debería haber 16 productos en recuadros 2) Dado que el paciente seleccione un producto Cuando el paciente seleccione un sitio o profesional Entonces el sistema debería permitir la relación entre el producto y la categoría seleccionada.		
Tareas <ol style="list-style-type: none">1. Configurar los modelos Producto, Sitio, y Profesional en Sequelize (1 hora).2. Crear la base de datos en MySQL y sincronizarla con Sequelize (1 hora).3. Crear endpoint para obtener la lista de productos en Express (1 hora).4. Crear endpoint para relacionar un producto con un sitio o profesional en Express (2 horas).5. Desarrollar la interfaz de visualización de productos en game.html (2 horas).6. Implementar la lógica en game.js para cargar y mostrar productos desde la API (1 hora).7. Implementar la selección de producto y asociación con sitio o profesional en game.js (2 horas).		

8. Realizar pruebas unitarias para los endpoints (1 hora). 9. Realizar pruebas de integración para verificar el flujo de selección y asociación (1 hora). 10. Realizar pruebas de interfaz de usuario en el navegador (1 hora).											
Check list:											
I	✓	N	✓	V	✓	E	✓	S	✓	T	✓

Nro: HU-002	Título: Relacionar profesional o sitio con productos	Prioridad: Alta Estimacion: 8 horas
Historia de usuario: Como Doctor quiero que las palabras de sitio y profesional sean recuadros que se puedan seleccionar para que el Paciente pueda relacionarlos con los productos.		
Criterios de aceptación: 1) Dado que un paciente esté jugando Cuando el paciente vea la lista de productos Entonces las categorías de sitio y profesional deberían estar organizadas en recuadros seleccionables para cada producto. 2) Dado que el paciente seleccione un recuadro Cuando el paciente relacione el sitio o el profesional con el producto Entonces la aplicación debería confirmar la selección con un cambio visual, como un borde resaltado.		
Tareas: 1. Definir el diseño de recuadros para las categorías "sitio" y "profesional" en el frontend (1 hora). 2. Implementar recuadros seleccionables para las categorías en la interfaz de juego usando HTML y CSS (2 horas). 3. Programar la lógica de selección de categorías con JavaScript (1 hora). 4. Establecer la funcionalidad de relación entre productos y categorías en el backend utilizando Node.js y Express (2 horas).		

<p>5. Implementar cambios visuales en el frontend para confirmar selección (e.g., borde resaltado) (1 hora).</p> <p>6. Pruebas de interfaz para confirmar el cambio visual y funcionalidad de relación (1 hora).</p>											
Check list:											
I	✓	N	✓	V	✓	E	✓	S	✓	T	✓

Segunda iteración:

En esta iteración se busca conseguir un agregado en cuanto a la interacción con el sistema, puesto que aparte de lo que se busca conseguir como doctor, se busca generar una funcionalidad en la que el usuario puede seguir mejorando constantemente su desempeño.

Nro: HU-003	Titulo: Notificar equivocación	Prioridad: Alta Estimacion: 10 horas
Historia de usuario: Como Doctor quiero que si el Paciente se equivoca en relacionar una categoría se le avise al paciente para que el paciente vea su error.		
Criterios de aceptación: 1) Dado que un paciente haya relacionado incorrectamente un sitio o profesional con un producto Cuando la relación sea incorrecta y termine toda las relaciones al dar click al botón “Finalizar” Entonces el sistema debe mostrar las equivocaciones y sus aciertos al paciente resaltándolos. 2) Dado que el paciente haya relacionado correctamente Cuando el paciente finalice el juego Entonces el sistema debería marcar toda sus respuestas correctas resaltándolas.		

Tareas:

1. Configurar en el backend la detección de relaciones correctas e incorrectas (2 horas).
2. Crear una funcionalidad que verifique todas las relaciones una vez que el paciente haga clic en “Finalizar” (2 horas).
3. Implementar en el frontend la visualización de resultados resaltados, diferenciando las respuestas correctas e incorrectas (2 horas).
4. Añadir un botón “Finalizar” en la interfaz de juego que invoque la función de verificación al ser presionado (1 hora).
5. Programar mensajes visuales para cada respuesta en el frontend, resaltando con colores los aciertos y errores (1 hora).
6. Realizar pruebas de interfaz y lógica para asegurar que las respuestas se marquen correctamente al finalizar el juego (1 hora).
7. Implementar pruebas de usabilidad para confirmar que el paciente pueda ver y entender sus errores y aciertos al finalizar (1 hora).

Check list:

I	✓	N	✓	V	✓	E	✓	S	✓	T	✓
---	---	---	---	---	---	---	---	---	---	---	---

Nro: HU-004	Título: Volver a jugar	Prioridad: Media Estimacion: 6 horas
Historia de usuario: Como Paciente, quiero tener la opción de volver a jugar después de finalizar una ronda, para intentar mejorar mi desempeño y reforzar mis conocimientos.		
Criterios de aceptación: 1) Dado que el paciente haya terminado una ronda del juego Cuando desee jugar de nuevo Entonces el sistema debe ofrecer una opción clara de "Volver a jugar" para iniciar una nueva partida desde el principio.		

2)

Dado que el paciente quiera reiniciar el juego durante una ronda en curso
Cuando elija la opción de "Volver a jugar"
Entonces el sistema recargara la página del juego

Tareas:

1. Diseñar e implementar un botón "Volver a jugar" en la interfaz de juego (1 hora).
2. Programar la funcionalidad en el backend para reiniciar el juego al seleccionar "Volver a jugar" (2 horas).
3. Configurar el reinicio de la partida y recarga de la pantalla de juego en el frontend (1 hora).
4. Crear pruebas de funcionalidad para confirmar que el botón "Volver a jugar" recarga la página y reinicia el juego correctamente (2 horas).

Check list:

I	✓	N	✓	V	✓	E	✓	S	✓	T	✓
---	---	---	---	---	---	---	---	---	---	---	---

Tercera iteración:

En esta iteración se busca mayor acercamiento al paciente, encontrando que el valor que se encuentra agregado radica en las funcionalidades que permiten al paciente mejorar en cuanto a su desempeño y, familiarizarse con el sistema de una forma mas amena y relativa en cuanto a lo que se tiene hasta el momento.

Nro: HU-005	Título: Volver al login	Prioridad: Media Estimacion: 4 horas
Historia de usuario: Como Paciente, quiero tener la opción de volver al inicio de sesión en cualquier momento, para salir de la sesión actual de forma rápida.		
Criterios de aceptación:		
1)		

<p>Dado que el paciente esté en cualquier parte del juego Cuando elija la opción de "Volver al login" Entonces el sistema debe cerrar la sesión actual y redirigir al paciente a la pantalla de inicio de sesión.</p>											
<p>Tareas:</p> <ol style="list-style-type: none"> 1. Añadir un botón "Volver al login" accesible en toda la interfaz de juego (1 hora). 2. Programar la lógica de cierre de sesión en el backend y redirección a la pantalla de inicio de sesión (1 hora). 3. Implementar la redirección en el frontend para regresar al login al hacer clic en el botón (1 hora). 4. Realizar pruebas de sesión y navegación para confirmar que el paciente es redirigido correctamente al login (1 hora). 											
Check list:											
I	✓	N	✓	V	✓	E	✓	S	✓	T	✓

Nro: HU-006	Título: Ver respuestas correctas	<p>Prioridad: Alta</p> <p>Estimacion: 6 horas</p>
<p>Historia de usuario: Como Paciente, quiero poder ver las respuestas correctas después de terminar el juego, para identificar mis errores y aprender de ellos.</p>		
<p>Criterios de aceptación:</p> <p>1)</p> <p>Dado que el paciente haya completado una ronda del juego Cuando desee revisar las respuestas correctas Entonces el sistema debe mostrar una lista o resaltado de las relaciones correctas para que el paciente pueda comparar con sus respuestas.</p>		
<p>Tareas:</p> <ol style="list-style-type: none"> 1. Implementar lógica en el backend para identificar y almacenar las relaciones correctas al finalizar el juego (2 horas). 		

2. Desarrollar en el frontend la interfaz que muestra las respuestas correctas o resaltado para cada relación (2 horas).											
3. Programar la funcionalidad de visualización de respuestas correctas en la pantalla de juego al finalizar una ronda (1 hora).											
4. Pruebas de interfaz y funcionalidad para asegurar que el paciente pueda ver y comparar sus respuestas correctamente (1 hora).											
Check list:											
I	✓	N	✓	V	✓	E	✓	S	✓	T	✓

Herramientas

Al referirse a las herramientas que se han utilizado para la realización de este release, se cuenta con:

Visual Studio Code (VS Code)

VS Code es el entorno de desarrollo integrado (IDE) utilizado para escribir y editar el código fuente del proyecto. Con varias extensiones, como aquellas para Node.js, Git y Docker, VS Code facilita el desarrollo, la depuración y la integración con herramientas adicionales.

Git y GitHub

Git es el sistema de control de versiones empleado para gestionar los cambios en el código y colaborar en el desarrollo. Permite rastrear el historial de cambios, crear ramas y gestionar el flujo de trabajo en equipo.

GitHub es el repositorio remoto donde se almacena el código fuente del proyecto. Además de ser una plataforma de colaboración, facilita el control de versiones, el despliegue y la documentación del proyecto.

Mermaid.js

Mermaid.js es una herramienta de creación de diagramas mediante código, que se usa para documentar visualmente la arquitectura del proyecto, los diagramas de entidad-relación, de clases y de flujo MVC. Mermaid facilita la creación de documentación clara y comprensible sin necesidad de herramientas gráficas adicionales.

Tecnologías

Node.js

Node.js es el entorno de ejecución de JavaScript en el servidor que se usa para el backend del proyecto. Permite manejar la lógica del negocio en el servidor, respondiendo a peticiones y gestionando el flujo de datos entre el frontend y la base de datos.

Express.js

Express.js es un framework para Node.js que simplifica la creación de APIs REST y la gestión de rutas. En este proyecto, Express se utiliza para definir y manejar las rutas del servidor, actuando como el controlador en la arquitectura MVC.

Sequelize

Sequelize es un ORM para Node.js que facilita la interacción con la base de datos MySQL. Permite definir modelos (entidades Producto, Sitio, y Profesional) y sus relaciones de manera sencilla, eliminando la necesidad de escribir consultas SQL manuales.

MySQL

MySQL es el sistema de gestión de bases de datos que almacena los datos persistentes del proyecto. Las entidades y relaciones se definen a través de Sequelize y se gestionan en MySQL, permitiendo consultas estructuradas y optimizadas para el sistema.

HTML, CSS y JavaScript

Estas tecnologías constituyen la base del frontend del proyecto:

- HTML estructura las vistas (index.html y game.html) y organiza el contenido.
- CSS define el estilo visual, proporcionando una interfaz atractiva y accesible para el usuario.

JavaScript maneja la lógica en el cliente, controlando la dinámica de la interfaz, especialmente en el archivo game.js, donde se cargan datos de la API y se interactúa con el DOM.

Arquitectura

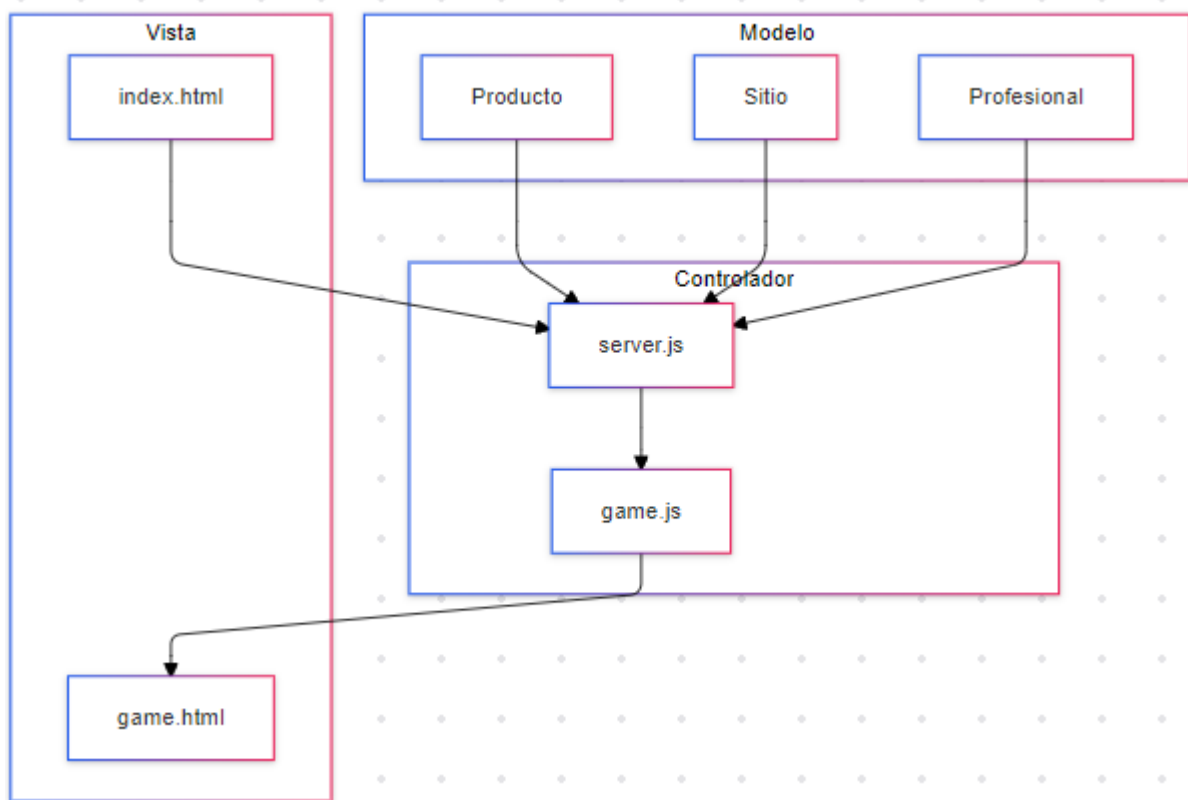
Diagrama de Arquitectura MVC

La arquitectura de nuestro software sigue el patrón MVC (Modelo-Vista-Controlador), que organiza la aplicación en tres componentes principales: modelo, vista y controlador. Esta arquitectura permite una separación clara de responsabilidades, facilitando el mantenimiento, la escalabilidad y la capacidad de extender la aplicación. A través de MVC, se logra una interacción fluida entre la lógica de datos, la interfaz de usuario y la lógica de control, lo que resulta en un flujo de datos eficiente y modular.

El diagrama de flujo muestra la arquitectura del proyecto basada en el patrón MVC (Modelo-Vista-Controlador), que organiza la aplicación en tres partes principales:

- **Modelo:** Incluye las clases Producto, Sitio y Profesional. Estas clases representan los datos del sistema y se comunican con el controlador del servidor (server.js) para devolver la información al cliente.
- **Controlador:** Se compone de server.js (controlador del servidor) y game.js (controlador del juego en el cliente). server.js gestiona las solicitudes desde la vista y obtiene datos del modelo. game.js manipula la interfaz de juego al obtener datos desde la API y actualizar el DOM de game.html.
- **Vista:** Incluye los archivos HTML (index.html y game.html) que representan la interfaz de usuario. index.html es la pantalla de login, mientras que game.html muestra el juego. Estos archivos HTML se comunican con los controladores para recibir y mostrar datos.

Este diagrama permite entender cómo fluyen los datos en la aplicación y cómo se comunican las distintas capas para presentar una arquitectura clara y modular.



Diseño

El diseño del producto se centra en la estructura interna de nuestro sistema, incluyendo la representación de los datos y la lógica del negocio. A través de diagramas detallados de entidad-relación y clases, se describe cómo se gestionan y estructuran las entidades clave, permitiendo un diseño orientado a objetos que facilita el desarrollo de funcionalidades y el manejo eficiente de los datos en toda la aplicación.

Diagrama Entidad-Relación (ER)

Este diagrama muestra la relación entre las entidades en la base de datos de nuestro proyecto.

Las entidades son Sitio, Profesional y Producto, y se han definido de la siguiente manera:

- **Producto:** Representa los diferentes productos en nuestro sistema y está asociado tanto a un sitio como a un profesional.

- Sitio: Representa los lugares donde se encuentran los productos (por ejemplo, tiendas específicas).
- Profesional: Indica a los profesionales responsables o relacionados con cada producto.

Las relaciones entre las entidades son:

- Cada Producto está relacionado con un solo Sitio y un solo Profesional.
- Sitio y Profesional tienen una relación de uno a muchos con Producto, indicando que cada sitio o profesional puede estar relacionado con múltiples productos.

Este diagrama ayuda a visualizar cómo los datos se organizan y conectan en el sistema, permitiéndonos comprender la estructura de la base de datos y las dependencias entre entidades.

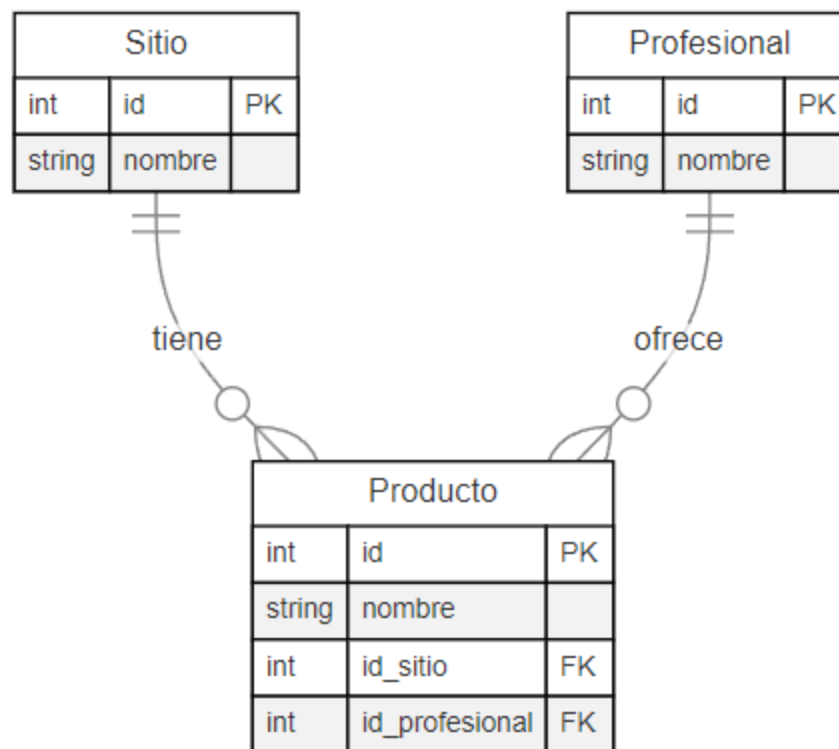
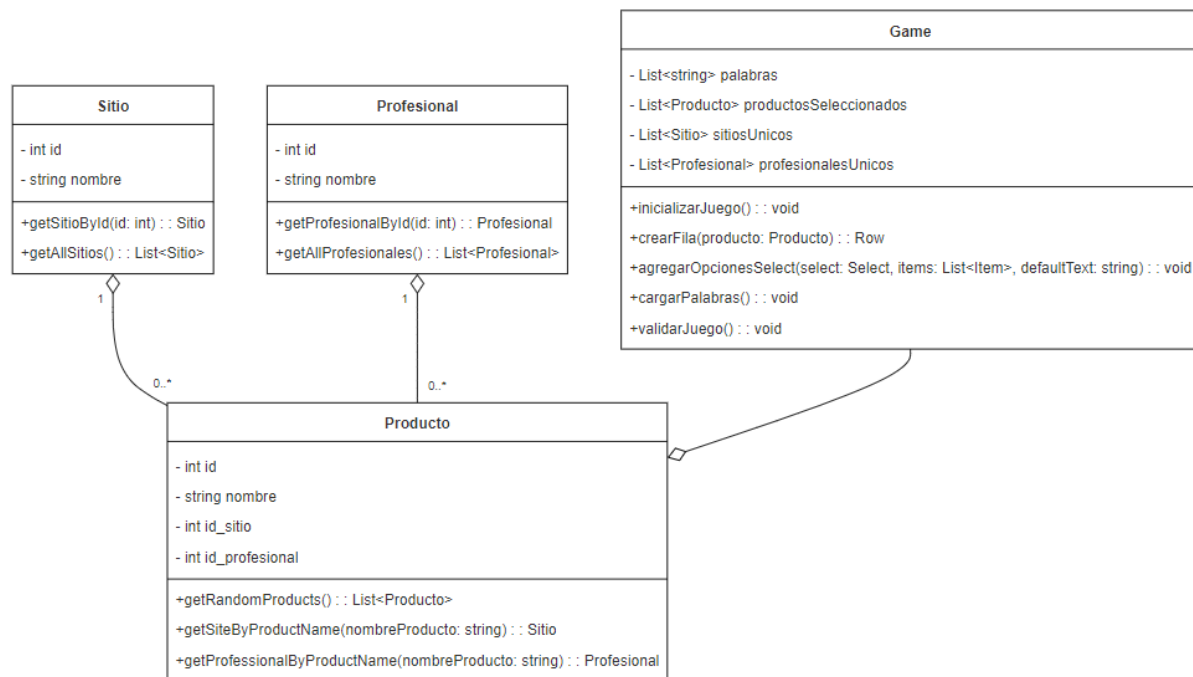


Diagrama de Clases

El diagrama de clases representa la estructura del código a nivel de clases y muestra las relaciones entre diferentes componentes de la aplicación. Para este proyecto, hemos creado clases principales que coinciden con nuestras entidades de base de datos, así como algunas clases adicionales que controlan la lógica de la aplicación:

- Sitio, Profesional, y Producto: Estas clases coinciden con las tablas en la base de datos y representan la capa del modelo en nuestro proyecto. Cada clase contiene atributos y métodos para acceder y manipular datos específicos de cada entidad.
- Game: Esta clase maneja la lógica del juego en el cliente. Contiene funciones para cargar datos, generar opciones de selección y validar el juego.
- Server: Este es el controlador del lado del servidor que maneja las peticiones API para acceder a los datos desde el cliente.

El diagrama de clases muestra cómo interactúan estos componentes, representando la lógica del negocio en el backend y las interacciones del cliente en el frontend.




Especificación de casos de prueba

Caso de Prueba: CP01

ID: CP01	Título: Visualización de productos en pantalla de juego.
-----------------	---


Descripción: Se espera probar que el sistema muestra correctamente los productos disponibles en la pantalla del juego cuando el paciente accede como invitado, validando que se visualicen 16 productos en recuadros.		
Procedimiento de prueba: <ol style="list-style-type: none"> 1. Ejecutar la aplicación y acceder a la pantalla de inicio de sesión. 2. Seleccionar la opción de "Invitado" para iniciar sesión. 3. Navegar hasta la pantalla del juego. 4. Observar la cantidad de productos visualizados en recuadros. 		
Resultados esperados: El sistema muestra exactamente 16 productos en recuadros, distribuidos en la pantalla del juego.		
EJECUCIONES DEL CASO DE PRUEBA		
Fecha: 06-11-2024 Responsable: William Enriquez	Resultados obtenidos: El sistema muestra los 16 productos correctamente en recuadros en la pantalla de juego.	Reprobado () Aprobado (X)

Capturas:

 Llave inglesa


Selecciona un sitio ▼

Selecciona un profesional ▼

 Esmeriladora


Selecciona un sitio ▼

Selecciona un profesional ▼

 Broca


Selecciona un sitio ▼

Selecciona un profesional ▼

 Sierra


Selecciona un sitio ▼

Selecciona un profesional ▼

 Cíncel


Selecciona un sitio ▼

Selecciona un profesional ▼

 Lijadora


Selecciona un sitio ▼

Selecciona un profesional ▼

 Cinta adhesiva


Selecciona un sitio ▼

Selecciona un profesional ▼

 Tornillo


Selecciona un sitio ▼

Selecciona un profesional ▼

 Taladro


Selecciona un sitio ▼

Selecciona un profesional ▼

 Soplete


Selecciona un sitio ▼

Selecciona un profesional ▼

 Sargento


Selecciona un sitio ▼

Selecciona un profesional ▼

 Destornillador


Selecciona un sitio ▼

Selecciona un profesional ▼

 Pistola de silicona


Selecciona un sitio ▼

Selecciona un profesional ▼

 Metro


Selecciona un sitio ▼

Selecciona un profesional ▼

 Alicata

Selecciona un sitio ▼

Selecciona un profesional ▼

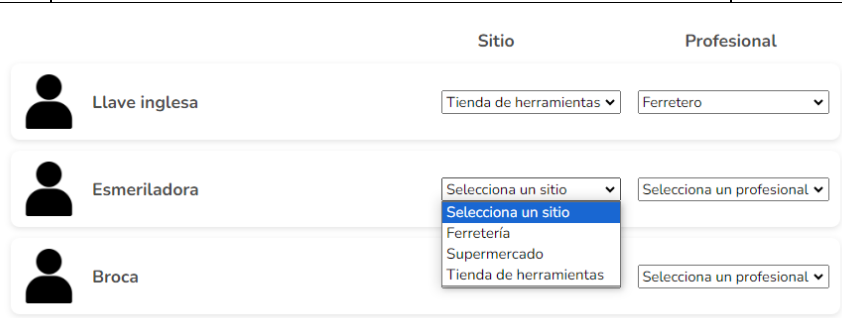
 Nivel

Selecciona un sitio ▼

Selecciona un profesional ▼

Caso de Prueba: CP02

ID: CP02	Título: Relación de producto con sitio/profesional
Descripción: Validar que el sistema permita la correcta asociación entre el producto seleccionado y la categoría elegida (sitio o profesional) cuando el paciente accede como invitado.	

Procedimiento de prueba: <ol style="list-style-type: none"> 1. Ejecutar la aplicación y acceder a la pantalla de inicio de sesión. 2. Seleccionar la opción de "Invitado" para iniciar sesión. 3. Navegar hasta la pantalla del juego y observar los productos. 4. Seleccionar un producto de la lista. 5. Asociar el producto a una categoría (sitio o profesional) disponible. 		
Resultados esperados: El sistema permite asociar el producto seleccionado a la categoría elegida (sitio o profesional) sin errores.		
EJECUCIONES DEL CASO DE PRUEBA		
Fecha: 06-11-2024 Responsable: Emilio Armas	Resultados obtenidos: La asociación entre el producto y la categoría se realizó correctamente.	Reprobado () Aprobado (X)
Capturas:		

Caso de Prueba: CP03

ID: CP03	Título: Marcado de respuestas correctas e incorrectas al finalizar el juego
Descripción: Este caso de prueba valida que el sistema, al finalizar una ronda del juego, marque las respuestas correctas en color verde y las incorrectas en color rojo, permitiendo al paciente distinguir claramente entre ambas para hacer una revisión efectiva de su desempeño.	

Procedimiento de prueba: <ol style="list-style-type: none"> 1. Ejecutar la aplicación e iniciar una ronda del juego. 2. Completar la ronda respondiendo a todas las preguntas. 3. Seleccionar la opción de finalizar el juego. 4. Observar el marcado de respuestas en pantalla: las respuestas correctas deberían aparecer en color verde y las incorrectas en color rojo. 		
Resultados esperados: El sistema marca las respuestas correctas en color verde y las incorrectas en color rojo, facilitando la revisión para el usuario.		
EJECUCIONES DEL CASO DE PRUEBA		
Fecha: 06-11-2024 Responsable: Emilio Armas	Resultados obtenidos: El sistema marca correctamente las respuestas: verde para correctas y rojo para incorrectas.	Reprobado () Aprobado (X)
Capturas:		

Juego de Selección de Palabras

	Sitio	Profesional
 Pan integral	Panadería ▼	Panadero ▼
 Rosquillas	Panadería ▼	Panadero ▼
 Cuaderno de cuadros	Tienda Escolar ▼	Tienda de utiles escolares ▼
 Costillas	Carnicería ▼	Carnicero ▼
 Maceta de girasoles	Floristería ▼	Florista ▼
 Tulipanes	Floristería ▼	Florista ▼
 Tijeras para el cabello	Peluquería ▼	Florista ▼
 Zapatos de niño	Tienda Escolar ▼	Florista ▼

Caso de Prueba: CP04

ID: CP03	Título: Visualización de respuestas correctas al finalizar el juego
Descripción:	

Probar que el sistema muestra las respuestas correctas al finalizar una ronda del juego, permitiendo al paciente comparar sus respuestas.		
Procedimiento de prueba: <ol style="list-style-type: none"> 1. Ejecutar la aplicación y completar una ronda del juego. 2. Seleccionar la opción para revisar las respuestas correctas al finalizar la ronda. 3. Observar la lista o resaltado de respuestas correctas en la pantalla. 		
Resultados esperados: El sistema muestra una lista o resalta las relaciones correctas de manera que el paciente pueda comparar con sus respuestas.		
EJECUCIONES DEL CASO DE PRUEBA		
Fecha: 07-11-2024 Responsable: William Enriquez	Resultados obtenidos: El sistema muestra las respuestas correctas de manera clara y comprensible..	Reprobado () Aprobado (X)

Título: Visualización de respuestas correctas al finalizar el juego.

Entrada:

- Paciente ha completado una ronda del juego.

Pasos:

- El paciente finaliza una ronda del juego.
- El paciente selecciona la opción para revisar las respuestas correctas.

Resultados esperados:

- El sistema muestra una lista o resalta las relaciones correctas de manera que el paciente pueda comparar con sus respuestas.

Condiciones:

El sistema debe mostrar las respuestas correctas de manera clara y comprensible para el paciente, permitiendo la comparación con sus respuestas.

Ejecución de la prueba:

Formulario de selección de sitio y profesional para la prueba. El formulario está dividido en tres secciones de selección, cada una con un icono de persona y un título. La primera sección es 'Metro' (rojo) con opciones 'Selecciona un sitio' y 'Selecciona un profesional'. La segunda sección es 'Nivel' (verde) con opciones 'Supermercado' y 'Vendedor'. La tercera sección es 'Destornillador' (verde) con opciones 'Ferretería' y 'Ferretero'. Debajo de estas secciones hay tres botones: 'Finalizar' (verde), 'Mostrar solución' (naranja, resaltado con un recuadro rojo) y 'Volver a Jugar' (azul).

Persona	Sitio	Profesional
Metro	Selecciona un sitio	Selecciona un profesional
Nivel	Supermercado	Vendedor
Destornillador	Ferretería	Ferretero

Finalizar

Mostrar solución

Volver a Jugar

Seleccionado se puede ver las respuestas:

Tabla de respuestas correctas. La tabla muestra la respuesta correcta para cada herramienta seleccionada. El encabezado de la tabla es 'Mostrar solución' (naranja). La tabla tiene tres columnas: Herramienta, Sitio y Profesional.

Mostrar solución		
Herramienta	Sitio	Profesional
Lijadora	Ferretería	Ferretero
Metro	Tienda de herramientas	Herrero
Punzón	Ferretería	Ferretero
Soplete	Tienda de herramientas	Herrero
Destornillador	Ferretería	Ferretero
Broca	Tienda de herramientas	Herrero
Cinzel	Tienda de herramientas	Herrero

Caso de Prueba: CP05

ID: CP05	Título: Opción de "Volver a jugar" después de completar una ronda	
Descripción: Probar que el sistema ofrece una opción clara y funcional de "Volver a jugar" una vez finalizada una ronda, permitiendo al paciente iniciar una nueva partida.		
Procedimiento de prueba: <div><div>1.</div>Ejecutar la aplicación y completar una ronda del juego.</div> <div><div>2.</div>Seleccionar la opción de "Volver a jugar".</div> <div><div>3.</div>Confirmar que se inicia una nueva partida desde el principio.</div>		
Resultados esperados: <p>El sistema muestra una opción clara de "Volver a jugar" y permite iniciar una nueva partida.</p>		
EJECUCIONES DEL CASO DE PRUEBA		
Fecha: 07-11-2024 Responsable: William Enriquez	Resultados obtenidos: La opción de "Volver a jugar" funciona correctamente y permite iniciar una nueva partida.	Reprobado () Aprobado (X)

Capturas:

Metro

Nivel

Destornillador

Finalizar

Mostrar solución

Volver a Jugar

Juego de Selección de Palabras

	Sitio	Profesional
Lijadora	<input type="text" value="Selecciona un sitio"/>	<input type="text" value="Selecciona un profesional"/>
Metro	<input type="text" value="Selecciona un sitio"/>	<input type="text" value="Selecciona un profesional"/>
Punzón	<input type="text" value="Selecciona un sitio"/>	<input type="text" value="Selecciona un profesional"/>
Soplete	<input type="text" value="Selecciona un sitio"/>	<input type="text" value="Selecciona un profesional"/>
Destornillador	<input type="text" value="Selecciona un sitio"/>	<input type="text" value="Selecciona un profesional"/>
Broca	<input type="text" value="Selecciona un sitio"/>	<input type="text" value="Selecciona un profesional"/>
Cíncel	<input type="text" value="Selecciona un sitio"/>	<input type="text" value="Selecciona un profesional"/>