



UNMDP - Facultad de Ingeniería - Departamento de Informática

Sistema de Mapas de Calor Aplicados a Movimientos Monetarios en Salas de Casino

Autor

Echeverría Luchini, Lautaro

Director

Genin, Fernando

Proyecto Final para optar al grado de Ingeniero en Informática

Mar del Plata, 2 de mayo del 2022



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



UNMDP - Facultad de Ingeniería - Departamento de Informática

Sistema de Mapas de Calor Aplicados a Movimientos Monetarios en Salas de Casino

Autor

Echeverría Luchini, Lautaro

Director

Genin, Fernando

Proyecto Final para optar al grado de Ingeniero en Informática

Mar del Plata, 2 de mayo del 2022

Indice de Contenidos

Agradecimientos	3
Resumen	4
Abreviaturas	5
1. Introducción	6
1.1 Fundamentación	8
1.2 Motivación	8
1.3 Objetivos	9
1.3.1 Objetivo General	9
1.3.2 Objetivos Específicos	9
1.4 Estructura del trabajo	10
2. Metodología	12
2.1 Enfoque del proyecto	12
2.2 Implementación	13
3. Aporte del Proyecto	14
3.1 Beneficiarios	14
3.2 Análisis FODA	14
3.3 Análisis de Viabilidad	15
3.4 Impacto	17
4. Marco Conceptual	18
4.1 Arquitectura Cliente / Servidor	18
4.2 Servicios web con API REST	19
4.2.1 Recursos y Accesos	20
4.3 Interfaz Drag and Drop	22
4.4 Mapa de Calor	22
5. Proceso de Desarrollo	24
5.1 Modelo del Proceso	24
5.2 Cronograma	26
5.3 Costo (Horas Hombre)	28

5.4 Riesgos del Proyecto	29
6. Producto	30
6.1 Análisis	30
6.1.1 Requerimientos	31
6.1.2 Requerimientos funcionales	31
6.1.3 Requerimientos no funcionales	32
6.1.4 Casos de Uso	33
6.2 Diseño	35
6.2.1 Diagrama de Despliegue	35
6.2.2 Diagrama de Arquitectura	36
6.2.3 Diagrama de Secuencia del Sistema	38
6.2.4 Interfaces de Usuario	39
6.3 Modelado de Base de Datos	44
6.4 Seguridad	46
7. Análisis de benchmarking	48
7.1 ¿Qué es Benchmarking?	48
7.2 Análisis de Competencia	49
7.3 Diferencias y oportunidades del sistema	51
8. Conclusiones Finales	52
8.1 Memorias del Proyecto	52
8.2 Cumplimiento y evolución de los objetivos	54
8.3 Planificación vs Ejecución	54
8.3.1 Desvíos	56
8.4 Obstáculos del proyecto	56
8.5 Aprendizaje	57
Bibliografía	58
Anexo I: Lenguajes y herramientas del sistema	61
Anexo II: Casos de Uso	68
Anexo III: Procesos operativos del sistema	84
Anexo IV: Manual de Usuario	86
Anexo V: Carta de conformidad del sistema	93

Agradecimientos

Agradezco, en primer lugar, a mis padres quienes me acompañaron durante el largo proceso universitario. Sin ellos, hoy no estaría en esta instancia final. Desde sus palabras de aliento, su constante apoyo, sus felicitaciones, sus alegrías en cada materia aprobada, y sus ilusiones y esperanzas depositadas en mí fueron el motor importante que me ayudó a seguir siempre con el objetivo claro y con las mismas ganas del primer día.

En segundo lugar, agradezco a mi novia, persona fundamental en los últimos años de la carrera y que me impulsó a más en todo momento, que me escuchó y acompañó en los momentos lindos y en los no tantos, donde me sostuvo emocionalmente para no bajar los brazos, quien se levantó temprano y se acostó tarde con tal de ser una compañía en esos días de estudio que solo hubiesen sido mucho más difíciles.

También agradezco a mi familia y amigos, por ser parte del proceso y ayudarme en lo necesario como también alentarme en cada momento vivido durante la carrera.

Gracias a mi director de proyecto Fernando Genin, por la predisposición en todo momento, por aceptar estar a cargo de mi proyecto final de carrera y por su ayuda en cada duda o inquietud que surgió durante un año de trabajo que duró el proyecto.

Por último, gracias a la Universidad Nacional de Mar del Plata y a todos sus profesores, por formarme académicamente con su excelente nivel y por hacerme crecer tanto profesionalmente como personalmente.

Resumen

El objetivo del Proyecto fue brindar una solución a la empresa Neogame S.A. que no poseía oportunamente. A partir de la necesidad planteada por la empresa, se propuso hacer un sistema de mapas de calor para observar cómo varía la ganancia u otros factores (ingresos o egresos de dinero) en las máquinas tragamonedas de las salas de casino.

La realización del trabajo se hizo acorde a los conceptos y etapas de un proceso de desarrollo de software en cascada. Se abordaron inicialmente los requerimientos de la empresa, continuando con las etapas de análisis, diseño, desarrollo, pruebas e implementación.

Los resultados obtenidos tuvieron un fuerte impacto en la operación diaria de la empresa, siendo un sistema aceptado y satisfactorio para los distintos roles dentro de la misma.

El proyecto fue una fuente de aprendizaje constante durante todo su desarrollo y etapas. Se adquirieron nuevos conceptos teóricos que fueron utilizados en la práctica. Además, se repasaron muchos otros aspectos vistos durante la carrera e integrados en este proyecto. Por ende, dejó una buena enseñanza y una reconfortante experiencia de poder aportar un sistema de gran utilidad a una empresa tan reconocida.

Abreviaturas

API - **A**pplication **P**rogramming **I**nterface - Interfaz de programación de aplicaciones

CSS - **C**ascading **S**tyle **S**heets - Hojas de estilo en cascada

DBMS/SGBD - **D**atabase **M**anagement **S**ystem - **S**istema de **G**estión de **B**ases de **D**atos

ERP - **E**nterprise **R**esource **P**lanning - Sistema de planificación de recursos empresariales

HTML - **H**yper**T**ext **M**arkup **L**anguage - Lenguaje de Marcas de Hipertexto

HTTP - **H**yper**T**ext **T**ransfer **P**rotocol - Protocolo de Transferencia de Hipertexto

ISP - **I**nternet **S**ervice **P**rovider - Proveedor de servicios de internet

JS - **J**ava**S**cript

REST - **R**epresentational **S**tate **T**ransfer - Transferencia de estado representacional

SQL - **S**tructured **Q**uery **L**anguage - Lenguaje de Consulta Estructurada

Capítulo 1

Introducción

Desde la década del 2006, la empresa Neogame S.A opera en la Argentina. Conformada por Newtronic S.A. y Kaskira S.A., Neogame ofrece un servicio integral de calidad y confort en materia de recreación (Casinos y Hotelería), respetando los valores históricos y culturales, cooperando con el desarrollo económico de cada región para así poder posicionar cada destino a nivel nacional e internacional.

En las provincias de Entre Ríos y Formosa desarrolla sus actividades principales y tiene presencia en otros lugares del país, contando con un aproximado de 6 salas de juego en cada provincia argentina, lo que lo ha transformado a lo largo de los años en el líder en las salas de casino.

Neogame S.A., a través de Newtronic S.A. es uno de los importadores clave de nuestro país, ingresando anualmente cientos de miles de dólares en artículos para juegos de casino, partes y accesorios a nuestro territorio aduanero desde Taiwán. En cada una de las salas de juego, la empresa posee entre 50 a 500 máquinas tragamonedas.

La máquina tragamonedas es un dispositivo de juego de azar que se ha originado desde un juego mecánico de tres rodillos y pagos estadísticamente simples, hasta convertirse en un conjunto complejo de sistemas digitales de diferentes juegos y pagos, eventos especiales e interfaces digitales y táctiles. Estas están sobrecargadas de animaciones y sonido envolventes, entre muchas otras ofertas de experiencia.

En la década del 80, y particularmente en la del 90, la máquina tragamonedas tuvo su mayor aceleración en cambios de aspectos tecnológicos, con la incorporación del microprocesador y la memoria de acceso aleatorio (RAM) para convertirse en un dispositivo totalmente digital.

Estas máquinas abandonaron progresivamente, la palanca tradicional que iniciaba el juego la que fue reemplazada por el botón spin; las monedas fueron reemplazadas por tickets (con el sistema TITO); los sonidos mecánicos y físicos, como las monedas golpeando la bandeja metálica, fueron reemplazados por parlantes y un banco de sonidos, los reel fueron reemplazados por una o más pantallas que proyectan, de forma digital, unos rodillos virtuales. Es decir, una máquina tragamonedas hoy, procesa, calcula y determina el juego íntegro desde el momento exacto en el que el jugador acciona (o toca) el botón spin.

En la actualidad, todo el análisis y explicación del sistema en una máquina de este tipo es complejo y errático (a propósito) y recae como objeto de estudio para disciplinas como la informática, la matemática y la estadística.

Las nuevas dinámicas del negocio del entretenimiento y más específicamente de los tragamonedas hacen necesarias nuevas herramientas que contribuyan a transparentar la gestión del juego y a su vez generen una fuente de información estratégico-comercial para saber cómo el público apostador utiliza los dispositivos.

Es en este sentido, que Neogame S.A., se planteó llevar adelante un control de gestión en las máquinas, analizando movimientos de ingreso y egreso de dinero (premios normales o jackpots) y un balance de ganancia total como diferencia entre ambos.

Esta necesidad dio origen a la idea de realizar un mapa de calor. De esta forma, las variables mencionadas se podrían visualizar de manera simple y gráfica en relación al estado de una o varias salas de juego.

1.1 Fundamentación

Un proyecto es “la búsqueda de una solución inteligente al planteamiento de un problema tendente a resolver, entre muchas, una necesidad humana” (Baca Urbina, 2001, p.2).

Por otro lado, Montealegre (2008) define un proyecto como “un conjunto articulado y coherente de actividades orientadas a alcanzar uno o varios objetivos siguiendo una metodología definida” (p.7).

En este proyecto se abordará el proceso de desarrollo del sistema con las características de mapas de calor, gráficos lineales, gráficos de barras y autogestión de salas mediante *drag and drop*. Por medio de estas características, la finalidad es proveer a la empresa Neogame S.A. de un sistema útil y funcional para todas las áreas que estén involucradas. También, como se detallará más adelante, será compatible para trabajar de forma interoperable con un sistema ERP que dispone la empresa.

1.2 Motivación

El desarrollo de un sistema de software siempre presenta desafíos que motivan a la hora de su realización. En este caso, la motivación personal fue poder elaborar el Proyecto Final de forma solitaria e integrar los conceptos aprendidos a lo largo de la

carrera unificados en una aplicación que cuente con un diseño de interfaces de usuario (frontend), pero también que funcione del lado del servidor mediante una API y una base de datos (backend). Por otra parte, la motivación empresarial fue buscar aportar a la empresa una herramienta nueva y novedosa, que le permita mejorar la calidad de la información recabada en sus máquinas tragamonedas de sus salas de juego en los distintos puntos del país donde desarrolla su negocio.

1.3 Objetivos

1.3.1 Objetivo General

El objetivo general es lograr una herramienta tecnológica que sirva como instrumento de gestión y medición de movimientos monetarios en máquinas de juegos de azar. Este objetivo no está centrado en obtener el mejor producto, sino en lograr, a través de una herramienta, una transformación digital. Un mapa de calor resulta la vía más sencilla para lograr esto.

1.3.2 Objetivos Específicos

- Confeccionar un mapa de calor para una sala elegida en un rango de fechas determinadas.
- Permitir múltiples versiones de mapas en relación a una variable de referencia elegida (ingreso, egreso, jackpot o ganancia). Por ejemplo: elegir la opción de ingresos que hubieron de Marzo a Mayo 2021.
- Lograr un comportamiento dinámico del sistema a través de la parametrización de su configuración.
- Realizar diversas capturas de datos (*snapshots*) sobre las salas.
- Obtener información cualitativa y cuantitativa, que permiten generar datos comparativos entre diferentes periodos tiempos.
- Analizar las versiones para predecir ventajas y desventajas.
- Eliminar o reducir las posibles pérdidas que existan.

- Realizar cambios que beneficien a la empresa, utilizando los recursos de manera eficiente.
- Graficar información de ganancia de una sala en un rango de fechas determinadas.
- Graficar información de ganancia de una máquina en un rango de fechas determinadas.

1.4 Estructura del trabajo

El **capítulo 2**, definirá la metodología del proyecto, teniendo en cuenta el enfoque del mismo y su implementación.

El **capítulo 3**, abordará los conceptos asociados al aporte del proyecto, destacando los beneficiarios, el análisis FODA, su viabilidad y el impacto en la empresa para la cual se desarrolló el sistema.

El **capítulo 4**, introducirá el marco conceptual, las definiciones y referencias que se utilizaron en el trabajo, qué es un mapa de calor y cuáles son sus beneficios.

El **capítulo 5**, tendrá como foco el proceso de desarrollo del sistema, los cronogramas planificados y ejecutados, el costo en horas estimadas y reales y los riesgos asociados al proyecto.

El **capítulo 6**, será el principal y se verá el ciclo completo de desarrollo del sistema del Proyecto Final, las diferentes etapas, el producto obtenido con sus interfaces de usuario y las funcionalidades que permite.

El **capítulo 7**, expondrá un análisis de benchmarking, donde se harán comparaciones entre sistemas similares y las ventajas que tiene el sistema en cuestión de este trabajo.

Por último, en el **capítulo 8**, se explicarán las memorias del proyecto, la planificación y la ejecución, los obstáculos durante el mismo, el cumplimiento de los objetivos y los

aprendizajes que dejó. Para esto, se hará una mirada retrospectiva desde el inicio del Proyecto Final.

Capítulo 2

Metodología

2.1 Enfoque del proyecto

Para llevar a cabo el Proyecto Final de carrera, se realizó una investigación exploratoria de las diversas tecnologías y lenguajes que permitan alcanzar los objetivos planteados.

Luego de una búsqueda inicial donde se priorizó un sistema de calidad y acorde a las necesidades de la empresa, se definió utilizar la puesta en práctica de diversos lenguajes y herramientas que colaboren en el desarrollo tales como: HTML, CSS, JS, JQuery y una API REST que se conecte a una base de datos relacional MySQL. (**Anexo I**)

Por otro lado, se sumó una biblioteca JS gratuita de código abierto para generar los gráficos solicitados por la empresa.

2.2 Implementación

La implementación del proyecto estuvo condicionada en sus primeras etapas por la mirada constructiva de quienes conforman Neogame, sufriendo varias redefiniciones hasta poder definir lo que sería el sistema para el Proyecto Final.

Si bien las versiones sobre el producto deseado fueron diversas en un comienzo, el alcance sufrió recurrentes modificaciones hasta llegar a su versión final, fruto de la adquisición de experiencia sobre la solución a desarrollar a través de algunas preferencias establecidas.

El sistema informático permitió operacionalizar y optimizar el tiempo de los empleados y la ganancia de la empresa.

Por supuesto, podría ser mejorable a futuro sumando nuevos elementos al sistema que aumenten los beneficios que pueda obtener la empresa del mismo.

Capítulo 3

Aporte del Proyecto

3.1 Beneficiarios

El beneficiario del Proyecto Final es la empresa Neogame S.A., la requirente del sistema y la que plantea la oportunidad en su negocio para llevar adelante el proyecto. La empresa en cuestión, también contará con el sistema para obtener provecho económico con su implementación.

3.2 Análisis FODA

El estudio de planeación estratégica tipo FODA (Fortalezas, Oportunidades, Debilidades, Amenazas) “es una herramienta que permite generar una imagen de la situación actual de una microempresa” (Trejo y Zuñiga, 2016, p.9).

La aplicación del análisis FODA, permite obtener un diagnóstico y tomar las medidas necesarias para que no afecten o lo hagan en la menor medida posible, el desarrollo de las actividades de la organización.

De acuerdo con Thompson y Strikland (1998):

Una vez realizado el análisis FODA, es posible actuar directamente sobre las fortalezas y las debilidades (factores internos) porque son variables que dependen de la organización. En cambio, las oportunidades y las amenazas son independientes (factores externos) y, por lo tanto, es complicado poder modificarlas.

El análisis FODA del Proyecto Final se muestra a continuación en la siguiente tabla:

Fortalezas	Debilidades
<ol style="list-style-type: none">1. Personal capacitado para utilizar el producto.2. Información actualizada de las máquinas.3. Motivación de la gerencia en usar el sistema.4. Buena infraestructura para trabajar.5. Comunicación y trabajo en equipo.	<ol style="list-style-type: none">1. Escasa documentación de procesos abarcados por el sistema.2. Poca información disponible de la empresa requirente.
Oportunidades	Amenazas
<ol style="list-style-type: none">1. Destacarse en el mercado como un producto innovador inexistente en la competencia.2. Comercializar el producto a otras empresas interesadas.3. Captar inversores de otros mercados que quieran usar el sistema.	<ol style="list-style-type: none">1. Pandemia y restricciones sociales.2. Menor concurrencia de las personas a salas de casino.3. Nuevos productos de juego que desplacen las máquinas tragamonedas.

3.3 Análisis de Viabilidad

El análisis de viabilidad tiene en cuenta todos los factores relevantes que afectan al proyecto, incluyendo las consideraciones económicas, técnicas, legales, planificación,

así como los estudios de mercado, para determinar la probabilidad de completar el proyecto con éxito.

El concepto de viabilidad de una propuesta de investigación, de manera general, responde a la pregunta: ¿el plan de trabajo propuesto permite alcanzar los objetivos con un riesgo razonable?

A continuación, se exponen tres de las variables que se tuvieron en cuenta a la hora de la evaluación del proyecto que determinaron su factibilidad y su posterior ejecución.

1) Técnica

El conocimiento en tecnología hace posible la concreción del producto deseado. La solución planteada se encuentra dentro de una solución demandada en el sector para conocer el flujo monetario en las máquinas ya descritas. El análisis y desarrollo tecnológico fácilmente se acopla a este requerimiento y llega a la meta deseada.

2) Económica

Para el desarrollo de este proyecto no se necesita ningún servicio o componente que requiera una inversión por parte del cliente. Todo el proceso de desarrollo, que en sí mismo es lo más costoso, es realizado mediante este proyecto. La fuente de información necesaria para el procesamiento de mapas de calor se encuentra disponible en Internet de manera libre y gratuita.

3) Temporal

Para el éxito del proyecto se tiene en cuenta un minucioso plan de trabajo, detallando las horas utilizadas por etapa y por mes para la consecución del mismo. Asimismo, se elabora un cronograma de proyecto. Se realizan primero las acciones técnicas para luego volcar en papel el desarrollo.

3.4 Impacto

Al hablar de impacto de un proyecto, se define como:

La contribución significativa de un proyecto al logro del Fin/Propósito, que es un problema sectorial, y es un cambio o conjunto de cambios duraderos en la sociedad, la economía, la ciencia, la tecnología y el medio ambiente que mejora o degrada sus indicadores, como resultado de la ejecución de acciones de Investigación-desarrollo-innovación implementadas en los marcos de la estructura organizacional de un proyecto, y su sinergia con otras contribuciones de proyectos u otras acciones de tipo administrativas, etc. (Rodríguez y Cobas, s.f.)

Realizando un análisis de las implicaciones que puede tener el sistema desarrollado para el Proyecto Final y su impacto empresarial, se resalta:

- Aumento de la productividad, obteniendo ganancias mayores.
- Mayor rendimiento laboral de los empleados, mejorando la velocidad en la operatoria diaria.
- Alta competitividad en el mercado, haciendo uso de una herramienta innovadora.
- Análisis organizacional de la situación de las máquinas.

Capítulo 4

Marco Conceptual

4.1 Arquitectura Cliente / Servidor

El modelo cliente/servidor es un “modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varios aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas” (Marini, 2012, p. 1).

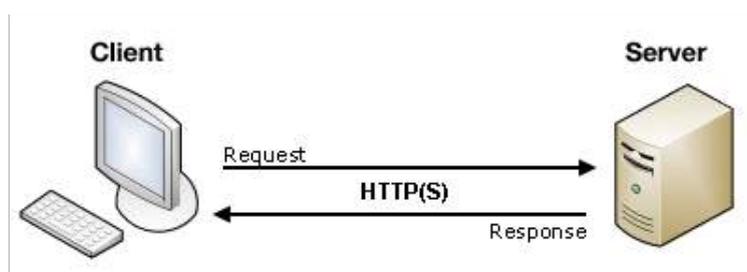


Gráfico 4.3.3: Esquema de modelo Cliente/Servidor

- **Cliente:** Programa ejecutable que participa activamente en el establecimiento de las conexiones. Envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo.
- **Servidor:** Es un programa que ofrece un servicio que se puede obtener en una red. Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante. Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación. El servidor comienza su ejecución antes de comenzar la interacción con el cliente.

La arquitectura cliente/servidor puede además tener multicapas. Por ejemplo, una capa de cliente que interactúe con el usuario final, una capa de servidores que trabaje con el almacenamiento en una base de datos para los servidores de aplicación y una capa intermedia de servidores de aplicación que procesa los datos para los clientes. Este trabajo, utiliza únicamente un modelo de dos capas (cliente y un servidor), como se muestra en la imagen 4.3.3.

4.2 Servicios web con API REST

Los servicios web son “aplicaciones auto-contenidas, auto-descriptivas y modulares, que pueden ser publicadas, localizadas e invocadas a través del web y que cuentan con un mecanismo estándar para establecer la comunicación con otros tipos de software a través de la red” (Arboleda, 2004, p.109).

En el año 2000, Roy Fielding definió en su tesis doctoral el término REST. Este sigue un conjunto de restricciones que cuando son aplicadas al diseño de un sistema, crean un estilo arquitectónico de software. Las características que debe cumplir una API para ser de tipo REST son:

- Debe ser un sistema cliente-servidor sin estado (cada petición al servicio tiene que ser independiente de las demás).
- Los datos pueden almacenarse en caché y optimizar las interacciones entre el cliente y el servidor.
- Debe ser un sistema uniformemente accesible (con una interfaz uniforme): cada recurso debe tener una única dirección y un punto válido de acceso. Los recursos se identifican con URIs, lo cual proporciona un espacio de direccionamiento global para el descubrimiento del servicio y de los recursos, que se acceden por peticiones HTTP (GET, POST, PUT, DELETE).
- Tiene que ser un sistema por capas: arquitectura jerárquica entre los componentes. Debe soportar escalabilidad.

Los servicios RESTful tienen múltiples ventajas que los destacan en el desarrollo de software: separación de un recurso de su representación, visibilidad, seguridad, escalabilidad y rendimiento.

4.2.1 Recursos y Accesos

Se denomina recurso REST a cualquier elemento direccionable a través de la web que puede ser accedido y transferido entre clientes y servidores.

Para acceder a los recursos de un sistema es necesario el protocolo HTTP. Este es un protocolo de transferencia de hipertextos. A lo largo de los años, se ha ido modificando y mejorando en sus versiones con el avance de la informática. Consta de diversos métodos de peticiones, entre los cuales, hay cuatro que son los más frecuentemente usados: GET, POST, PUT y DELETE.

Los métodos de peticiones HTTP son los siguientes:

- GET: El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

- POST: El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.
- PUT: El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.
- DELETE: El método DELETE borra un recurso en específico.
- HEAD: El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.
- CONNECT: El método CONNECT establece un túnel hacia el servidor identificado por el recurso.
- OPTIONS: El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino.
- TRACE: El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.
- PATCH: El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

Por medio de estos métodos, el sistema web del cliente podrá recibir o enviar información, según el tipo de petición, al servidor que establezca la conexión. En este trabajo, se utilizaron peticiones GET y POST, ya que fue necesario obtener la información para las distintas funcionalidades y además modificarla en el servidor.

El protocolo HTTP funciona sobre el protocolo ya mencionado TCP. Su principal ventaja es la transmisión fiable de datos de extremo a extremo. Por consiguiente, es el elegido por HTTP, ya que obliga establecer una conexión previa entre las dos máquinas antes de poder transmitir algún dato y la información que envía el emisor llega de forma correcta al destino.

4.3 Interfaz Drag and Drop

El concepto *Drag & Drop* (arrastrar y soltar en español), es una forma de interacción del usuario con el sistema, donde como su nombre lo indica, puede arrastrar y soltar elementos en un área específica llamada *Drop Zone*.

Existen muchas formas de aplicar *Drag & Drop* en un sistema: agregar imágenes, documentos, formularios, botones, encabezados, bordes, cambiar el color o tamaño de las fuentes, entre otras. También, existen lenguajes de programación orientados al *Drag & Drop* como es el caso de Scratch, un lenguaje creado por Grupo Lifelong Kindergarten del MIT Media Lab. Su principal objetivo, es el aprendizaje de programación sin tener conocimientos previos y, por ende, de gran utilidad para personas que están empezando a programar.

El sistema desarrollado en este trabajo, emplea el concepto de arrastrar y soltar, para poder agregar, eliminar y seleccionar máquinas dentro de una sala específica para moverlas libremente. De ese modo, permite una interfaz fácil de usar y una alta flexibilidad en la disposición de las máquinas dentro de la sala.

4.4 Mapa de Calor

Los mapas de calor son representaciones gráficas que sirven para observar los resultados de mediciones obtenidas reales mediante zonas de diferentes colores. La representación en estos mapas es por medio de una termografía donde los colores son más cálidos (rojo, naranja, amarillo) para determinados valores y más fríos para otros (azul, verde). Es una técnica muy útil y rápida para hacer un análisis del desempeño una zona seleccionada como parámetro.

Los objetivos perseguidos por los mapas de calor, varían de acuerdo a las mediciones llevadas a cabo. En los departamentos de marketing, son utilizados para medir

habitualmente los comportamientos de los usuarios en las páginas web. Algunos de los mapas de calor que se aplican son:

- Mapas de calor basados en clicks.
- Mapas de calor basados en el movimiento del mouse.
- Mapas de calor basados en scroll.

Para el proyecto, los mapas de calor se formaron con los resultados del juego de los usuarios en las máquinas tragamonedas de las distintas salas.

Capítulo 5

Proceso de Desarrollo

“Un proceso es una colección de actividades de trabajo, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo” (Pressman, 2010, p.27).

Por otro lado, hay autores que resaltan que “Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Captura y presenta las mejores prácticas que el estado actual de la tecnología permite. En consecuencia, reduce el riesgo y hace el proyecto más predecible” (Booch, Jacobson y Rumbaugh, 2018, p.4).

En el presente capítulo, se detallarán las etapas del proceso de desarrollo del Proyecto Final, las horas implementadas y los riesgos asociados en su elaboración.

5.1 Modelo del Proceso

Los modelos de desarrollo de software ofrecen un marco de trabajo usado para controlar el proceso de desarrollo de sistemas de información. Estos modelos permiten

reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar.

El modelo utilizado en el Proyecto Final fue un modelo de desarrollo en cascada. La decisión de su implementación se debió a las siguientes ventajas que presenta:

- Permite una buena estimación de los tiempos de entrega finales.
- Es fácil de manejar los planes de proyectos.
- Es un modelo lineal simple de implementar.

Otro factor importante para adoptar el modelo en cascada, fueron los requerimientos bien definidos inicialmente, gracias a que estos se iban a mantener en el tiempo y durante el proceso de desarrollo.

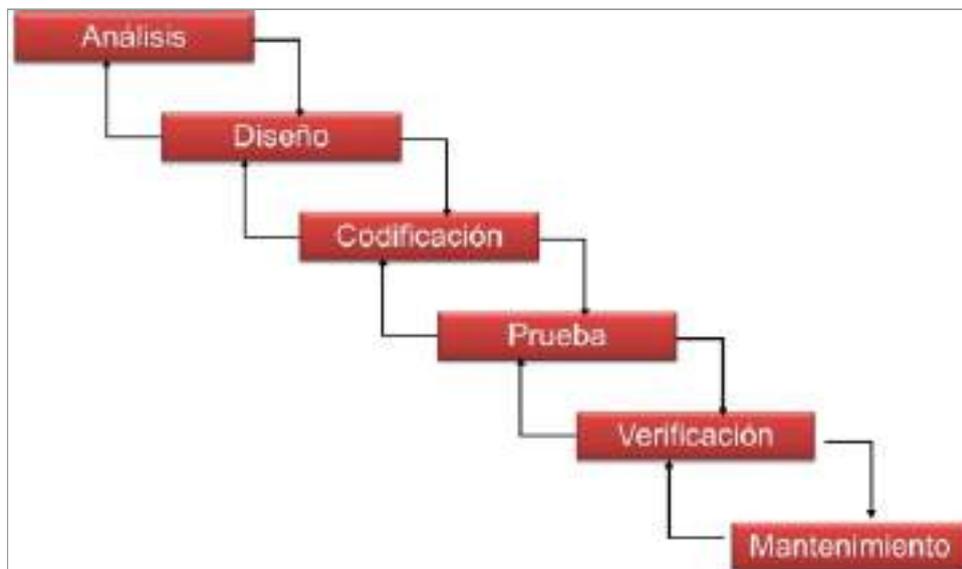


Gráfico 5.1: Etapas del Modelo de Desarrollo en Cascada

En lo personal la decisión de usar el modelo en cascada se debió a la experiencia sobre el mismo, siendo utilizado en numerosas asignaturas de la carrera y, en consecuencia, el modelo más familiarizado para desarrollar un sistema frente a los otros modelos del mercado.

El resultado de haber adoptado el modelo en cascada, no fue favorable respecto a los plazos estimados y en una mirada hacia atrás en el tiempo, no se volvería a elegir usar dicho modelo. Uno de los mayores problemas identificados por usar el modelo en cascada, fue la modificación de ciertos requerimientos con el desarrollo del sistema avanzado en gran medida. Esto retrasó la finalización de la etapa de desarrollo y reabrió las pruebas unitarias en los componentes finalizados.

5.2 Cronograma

El proyecto tuvo una estimación definida en el protocolo de aprobación del mismo. Se confeccionó un diagrama de Gantt estimado (gráfico 5.2.1) con el tiempo de cada etapa del desarrollo del sistema.



Gráfico 5.2.1: Diagrama de Gantt estimado Proyecto Final

Las tareas indicadas en el diagrama son las siguientes:

- 1) Identificación de problemas, oportunidades y objetivos.
- 2) Determinación de los requerimientos humanos del sistema.
- 3) Análisis de las necesidades del sistema.
- 4) Diseño y modelado del sistema.
- 5) Desarrollo del sistema.
- 6) Testing del sistema.
- 7) Documentación del sistema.



Gráfico 5.2.2: Diagrama de Gantt ejecutado Proyecto Final

En el diagrama de Gantt ejecutado (gráfico 5.2.2), se puede observar que las tareas 3), 4), 5), 6) y 7) fueron mayores a los plazos planificados previo a comenzar el proyecto. Luego de analizar los diferentes factores desencadenantes, se llegó a las siguientes conclusiones:

- En las etapas de análisis y diseño, en la primera etapa, hubo una mala estimación de horas, menos disponibilidad horaria de la prevista, requerimientos no claros e incertidumbre en aspectos del diseño del sistema a desarrollar.
- Las etapas de desarrollo y testing, en la primera etapa, fueron bien estimadas y no hubo diferencias notables en los plazos.
- En el mes de noviembre, luego de reuniones con el referente de la empresa Pablo Safronchik, surgieron nuevos requerimientos con el sistema avanzado. Esto derivó en nuevas etapas de análisis y diseño para poder continuar, generando más horas de las planeadas y nuevas funcionalidades a desarrollar.
- En concordancia con el punto anterior, las etapas de desarrollo y testing que estaban finalizando en una primera etapa, fueron abordadas nuevamente para cumplir con los nuevos requerimientos vistos.
- La etapa de documentación fue mucho mayor a lo esperado. Se invirtieron más horas que las planificadas en los mismos meses estimados y además, se hizo más extensa dando como resultado su culminación en una fecha posterior. Como se verá más adelante, fue una etapa muy compleja por los escasos recursos personales de escritura y redacción.

5.3 Costo (Horas Hombre)

El costo en horas hombre planificado al comenzar, fue equitativo para la mayoría de las etapas, destacando las etapas de desarrollo y testing como las más costosas. En el gráfico 5.3.1 se puede ver la distribución en porcentajes de cada etapa del proyecto.



Gráfico 5.3.1: Horas Planificadas del Proyecto Final

En el gráfico 5.3.2 se puede observar la distribución en porcentajes de las horas ejecutadas reales del proyecto. Al comparar ambos gráficos, se puede contemplar que todas las etapas del proyecto excepto la de documentación fueron menores a las planificadas. Se concluye que esta observación se debe a la diferencia significativa en las horas totales de la etapa de documentación siendo retrasada al doble del tiempo proyectado inicialmente, aumentando su porcentaje con respecto a los demás del gráfico.



Gráfico 5.3.2: Horas Ejecutadas del Proyecto Final

5.4 Riesgos del Proyecto

Según PMBOK (2016), un riesgo es un “Evento o condición incierta que, de ocurrir, tiene un efecto positivo o negativo en al menos uno de los objetivos del proyecto, tal como tiempo, costo, alcance o calidad”.

En el proceso de desarrollo de software, para asegurar un producto de calidad es necesario hacer un estudio previo de los riesgos que podría sufrir dicho producto, ya sea en su funcionamiento o en las etapas previas que interfieran su alcance.

Los riesgos identificados al sistema desarrollado para el Proyecto fueron:

Nro. Riesgo	Riesgo identificado	Consecuencia
01	Plan de desarrollo optimista	No cumplir con fecha de entrega final del proyecto
02	Relevamiento incorrecto de los requerimientos	Un sistema parcialmente funcional
03	Diseño de interfaces no intuitivo	Experiencia de usuario insatisfactoria
04	Curva de aprendizaje del sistema extensa	Soporte constante y mal uso del sistema

Capítulo 6

Producto

Un producto de software es un programa informático que resuelve una necesidad comercial de una empresa. Para obtener el producto, es preciso seguir las etapas correspondientes al modelo de proceso de desarrollo elegido. Como se mencionó en el capítulo 5 (Proceso de Desarrollo), el modelo utilizado en este proyecto, es el modelo en cascada. En las próximas secciones del capítulo, se mostrarán en detalle las etapas del modelo y sus características.

6.1 Análisis

La etapa de análisis se enfoca en definir qué es lo que se necesita al hablar del producto a desarrollar. Se definen los requerimientos, se evalúa su factibilidad, la descripción de los datos, las funciones que debe hacer el software y el comportamiento que debe tener. Es una etapa fundamental en el desarrollo, ya que una definición correcta y precisa de lo que requiere el usuario, dará como resultado un producto que satisfaga sus expectativas.

En el proyecto, la etapa de análisis consistió en definir en primera instancia el tipo de sistema que buscaba la empresa Neogame S.A. Una vez acordado en desarrollar un sistema web que corriera en cualquier sistema operativo, se formularon los requerimientos del sistema y sus funcionalidades. En las secciones 6.1.2 y 6.1.3 se detallarán los mismos.

La comunicación entre las partes para identificar los requerimientos del sistema y los objetivos fue siempre entre el director del proyecto Fernando Genin y el referente funcional y gerente de sistemas de la empresa Pablo Safronchik. Se realizaron reuniones virtuales y se intercambiaron mails para el relevamiento de los requerimientos. En el transcurso de su desarrollo, también se fueron explicando y mostrando en un servidor de prueba los nuevos avances.

6.1.1 Requerimientos

Los requerimientos de un sistema son “descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones de su operación” (Sommerville, 2005, p.83). Estos se dividen en dos tipos: requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales describen lo que el sistema debe hacer y las respuestas que dará a diferentes eventos de entrada que se presenten en su operación. En cambio, los requerimientos no funcionales son restricciones de características o propiedades que afectan a todo el sistema.

6.1.2 Requerimientos funcionales

- Será posible ingresar al sistema mediante un proceso de identificación y autenticación de usuarios.
- El sistema deberá contar con distintos roles de usuario y permisos según sea necesario.
- El sistema deberá permitir seleccionar una sala, un sector y fechas asociadas a un período para crear un mapa con dichos parámetros.

- El sistema deberá permitir agregar y eliminar máquinas dentro de un mapa.
- El sistema deberá permitir mover las máquinas libremente en el mapa.
- El sistema deberá mostrar datos estadísticos del mapa en el periodo de fechas asociadas.
- El sistema mostrará una tabla con los resultados de todas las máquinas en la sala, sector y fechas seleccionadas donde habrá información de ingresos, egresos, jackpots, ganancias, valor de créditos y juegos de las máquinas.
- El sistema deberá permitir hacer consultas de fechas diferentes a las seleccionadas en la creación del mapa, cambiando los colores del mapa en caso de haber variaciones monetarias.
- El sistema deberá permitir que un mapa sea filtrable según dinero ingresado, dinero extraído, jackpots o ganancias modificando las leyendas asociadas.
- El sistema deberá contar con leyendas parametrizables para ser mantenible en el tiempo.
- Los colores de las máquinas en los mapas, indicarán valores promedios en base al total acumulado dividido los días operativos.
- El sistema deberá permitir mostrar, guardar y eliminar versiones de mapas almacenadas.
- El sistema contará con gráficos estadísticos de ganancia de las salas y sus máquinas filtrables por fecha.
- El sistema deberá permitir ver las salas de casinos de la empresa.
- El sistema deberá permitir visualizar los usuarios registrados.

6.1.3 Requerimientos no funcionales

- El sistema será un sistema web y accesible desde un navegador.
- El sistema deberá tener conexión a Internet para su usabilidad.
- El sistema será independiente del sistema operativo utilizado.
- El sistema deberá ser interoperable con el sistema ERP de la empresa.
- El sistema deberá ser dinámico y sostenible en el tiempo, siendo actualizado con los nuevos resultados diarios de las máquinas.

- Los usuarios del sistema no se podrán registrar en el mismo, sino que serán dados de alta por la gerencia.
- El sistema proporcionará mensajes de error cuando la información ingresada sea incorrecta.
- El tiempo de aprendizaje del sistema, no deberá ser mayor a una jornada laboral diaria de 8 horas.

6.1.4 Casos de Uso

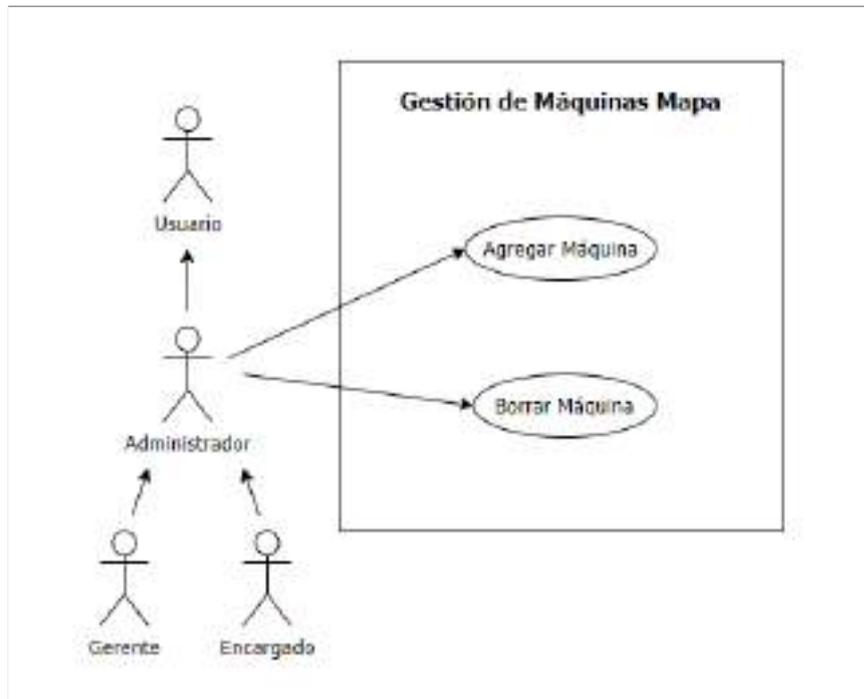
Los casos de uso cuentan con las siguientes características: nombre de caso de uso, actores, propósito, descripción, precondiciones, curso normal de los eventos y cursos alternos. En el **Anexo II** se detallan los casos de uso del sistema.

En base a los requerimientos funcionales identificados, los diagramas de casos de uso empleados para las funcionalidades del mapa son:

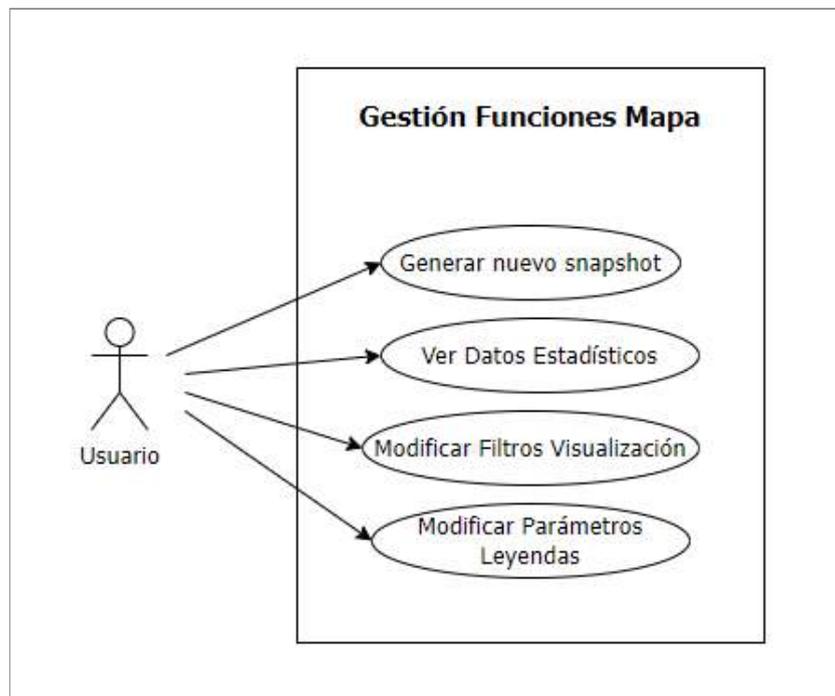
- Versiones de mapa:



- Máquinas de mapa:



- Operaciones del mapa:



6.2 Diseño

La etapa de diseño es otra de las etapas principales en la ingeniería de software para el desarrollo de sistemas. Comienza una vez que se han analizado y modelados los requerimientos. Además, es la última etapa antes de la codificación y sienta las bases para un producto de alta calidad.

Como menciona Pressman (2010, p.185), “sin diseño se corre el riesgo de obtener un sistema inestable, que falle cuando se hagan cambios pequeños, o uno que sea difícil de someter a prueba, o en el que no sea posible evaluar la calidad hasta que sea demasiado tarde en el proceso de software, cuando no queda mucho tiempo y ya se ha gastado mucho dinero”. Por lo tanto, es una etapa fundamental en el proceso.

6.2.1 Diagrama de Despliegue

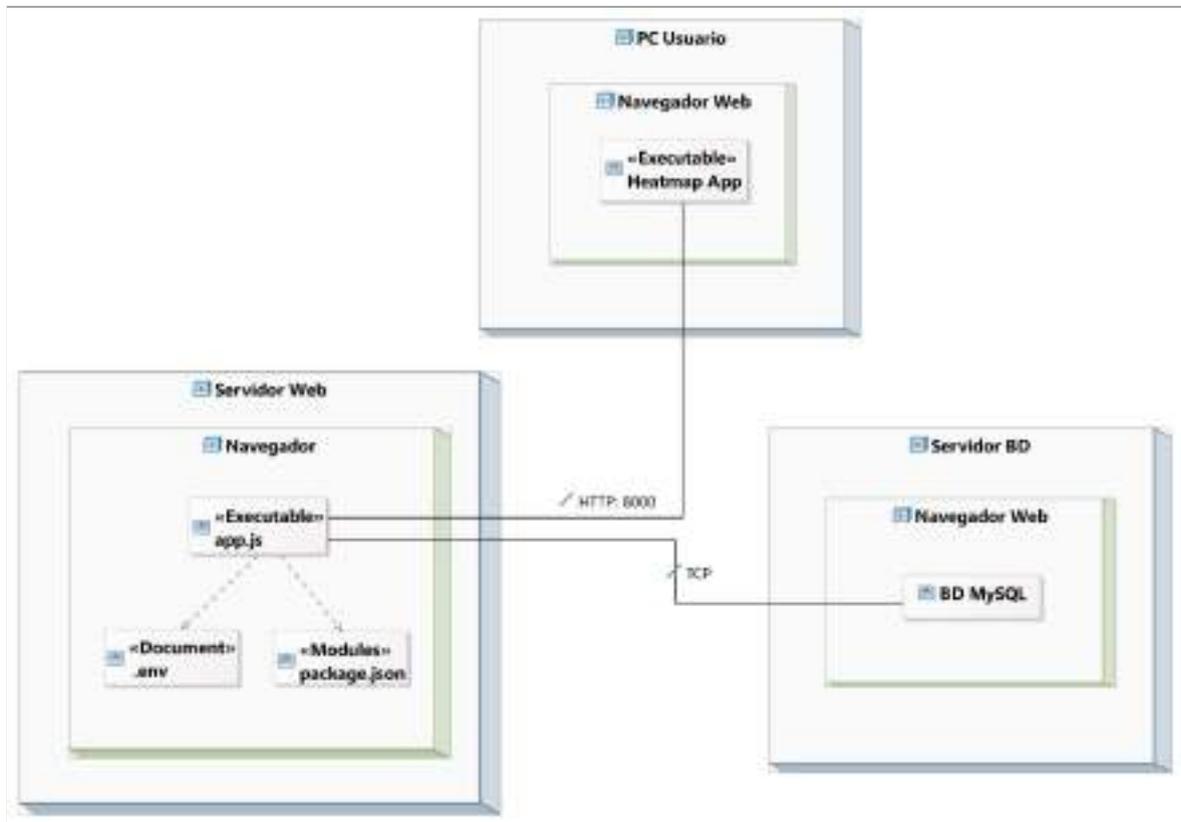
Un diagrama de despliegue es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema físico.

Los componentes de un diagrama de despliegue son:

- **Nodo:** es un recurso computacional sobre el cual se pueden desplegar artefactos.
- **Dispositivo:** representan máquinas físicas.
- **Ambiente de ejecución:** software que las aplicaciones requieren en tiempo de ejecución.
- **Artefacto:** representa un elemento de información que se usa o produce mediante un proceso de desarrollo de software o mediante la operación de un sistema.

- Asociación: una ruta de comunicación entre los nodos.

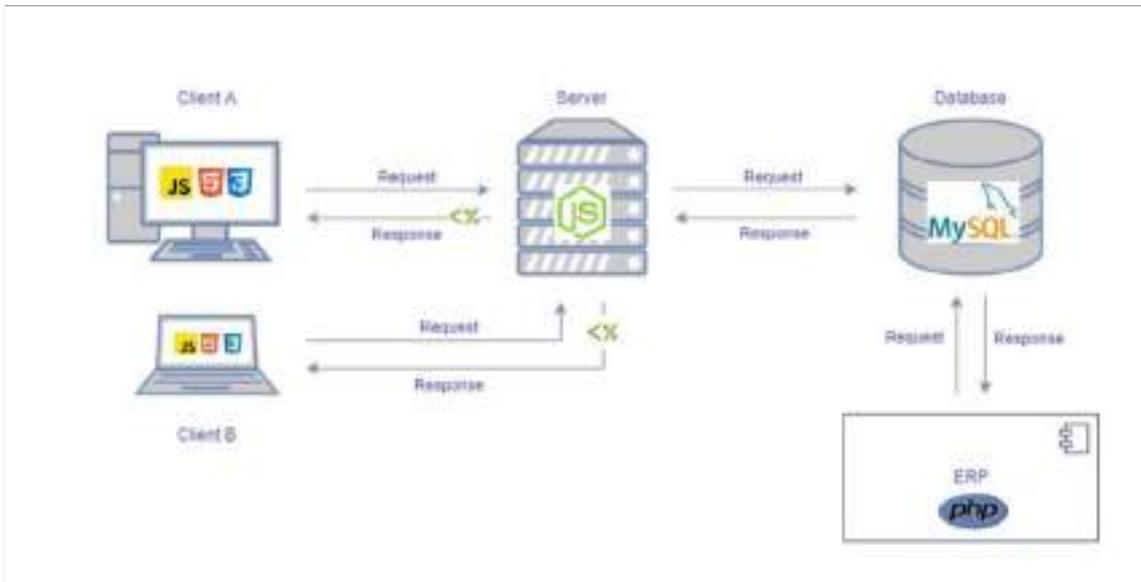
El diagrama de despliegue diseñado para el sistema es el siguiente:



6.2.2 Diagrama de Arquitectura

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos.

El diagrama próximo ayuda a comprender la arquitectura del sistema desarrollado en este proyecto:



Los conceptos y procesos que se muestran en el diagrama son:

- El sistema ERP hecho en **PHP** almacena o consulta datos de las máquinas tragamonedas en la base de datos **MySQL**.
- Los clientes del sistema desarrollado solicitan al servidor **Node.js** utilizando **JavaScript** y **AJAX** las vistas del sistema.
- El servidor recibe solicitudes HTTP y devuelve las plantillas **EJS** a los clientes.
- Los clientes reciben la respuesta del servidor junto a las vistas correspondientes **EJS** y se renderizan en formato **HTML** con estilos **CSS**. Las funcionalidades y acciones de las vistas son a través de **JavaScript**.
- Los clientes realizan modificaciones o quieren visualizar información (mapas, gráficos, tablas) para lo cual consultan al servidor por peticiones HTTP.
- El servidor se conecta con la base de datos **MySQL** compartida con el sistema ERP, obtiene los datos actualizados al momento y reenvía la información a los clientes que la solicitaron.

Las tecnologías y módulos que utiliza cada componente son:

- Cliente: HTML5, CSS3, JavaScript, JQuery, AJAX, Chart.js.
- Servidor: Node.js, Express.js, CORS, Dotsenv, EJS, Express-Session, Helmet, MD5, Nodemon.

- Base de datos: MySQL.
- Sistema de gestión ERP: PHP.

La elección de las tecnologías fue por dos factores principales: el primero, conocimiento y experiencia en las mismas, y el segundo, por la amplia documentación, soporte y tendencia en su utilización. Por otro lado, no se escogió el uso de los frameworks populares de JavaScript (Angular, React o Vue.js) para la aplicación del lado cliente dado que, no se tenía conocimiento en ellos, lo cual representaba una inversión de tiempo en formación previo a poder aplicarlos en el desarrollo del sistema.

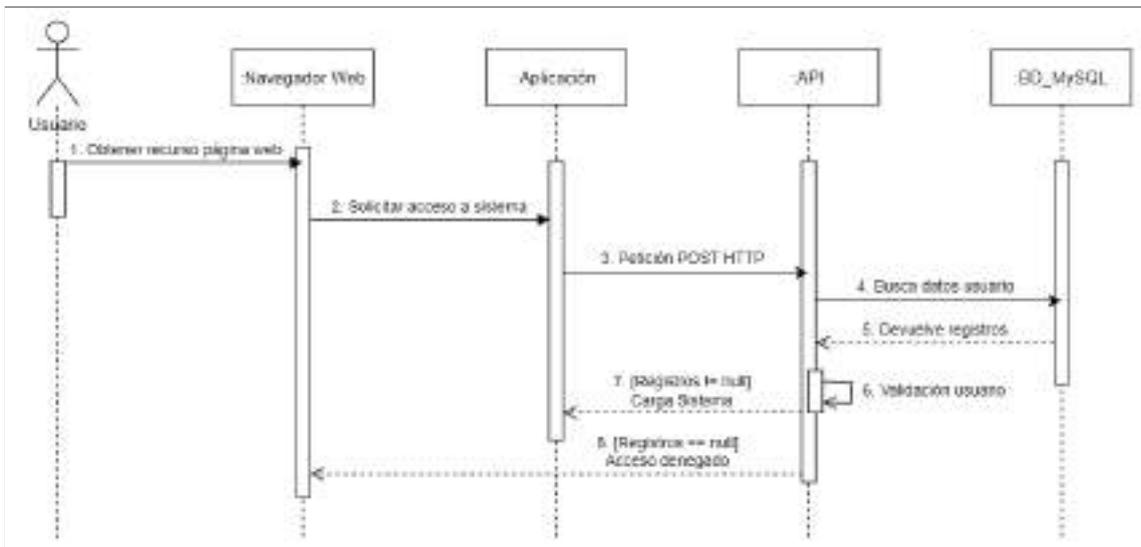
6.2.3 Diagrama de Secuencia del Sistema

Un diagrama de secuencia es un diagrama de UML que representa el orden de los eventos de manera cronológica en un sistema. Describe cómo se da la comunicación entre las partes del sistema, las solicitudes y las respuestas que se generan en cada una. Un diagrama de secuencia muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo.

Los elementos principales de un diagrama de secuencia son:

- Línea de vida: representa el tiempo en una línea vertical hacia abajo.
- Objeto: representa una clase u objeto UML del sistema.
- Actor: representa un usuario.
- Mensaje: representa una comunicación en el tiempo entre actores, objetos y las líneas de vida.

Se muestra a continuación, el diagrama de secuencia básico del sistema:

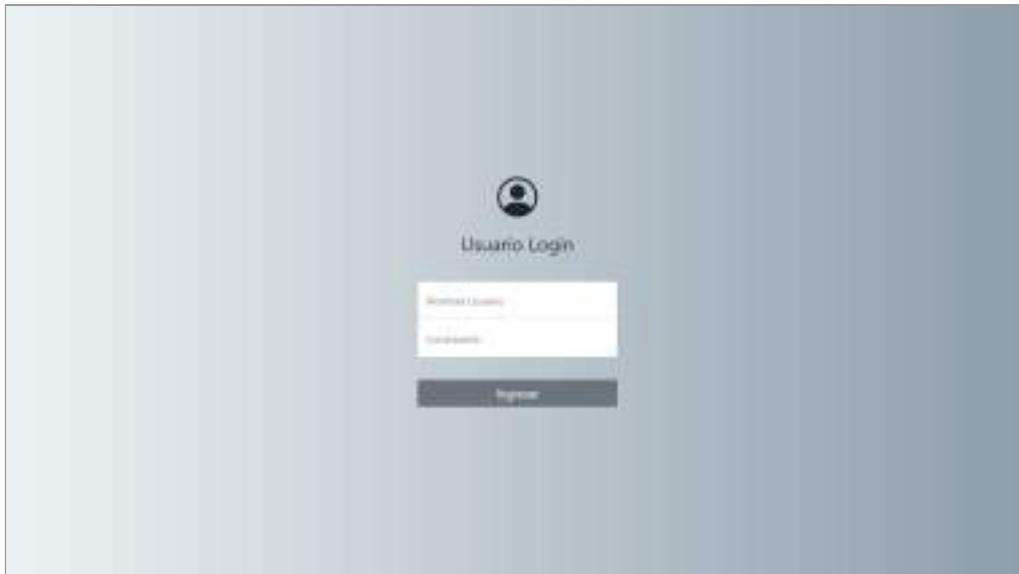


6.2.4 Interfaces de Usuario

Las interfaces de usuario de un sistema son las pantallas de visualización, interacción y comunicación de un humano con una máquina. Son las responsables de la experiencia del usuario (humano) al usar un sistema. Por ende, es importante un buen diseño teniendo en cuenta las necesidades del usuario y la distribución visual de las funcionalidades y requisitos detallados en la etapa de análisis.

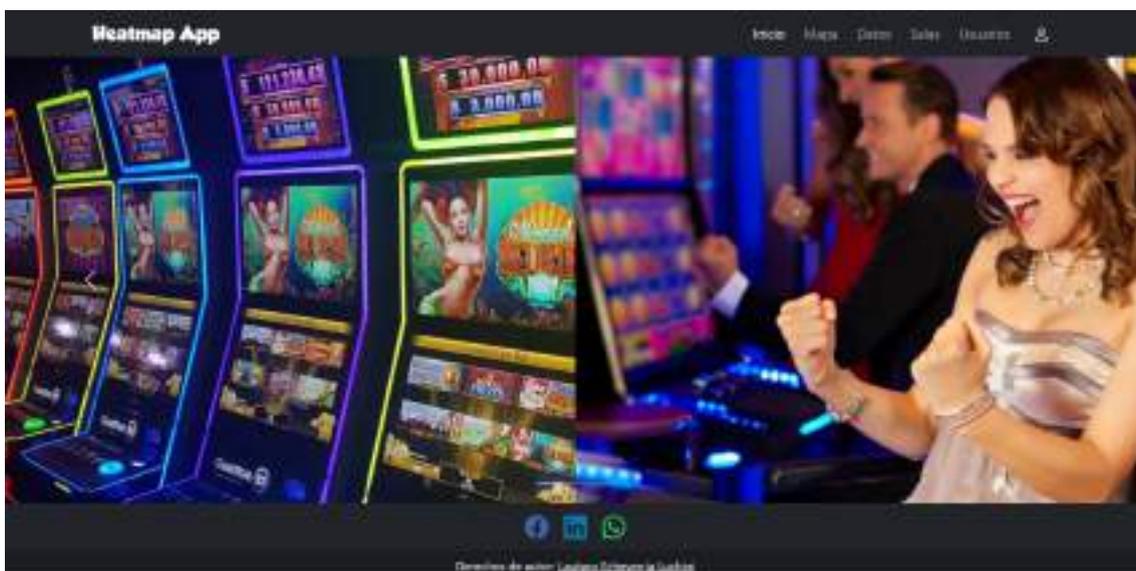
Las interfaces diseñadas para el Proyecto Final son:

- Login:



Es la interfaz de ingreso del usuario al sistema, que cuenta con un formulario de identificación y autenticación. Una vez ingresado al sistema, se crea una variable de sesión para el usuario. En caso contrario, el intento de acceso a otra pantalla del sistema será redirigido a esta.

- Inicio:



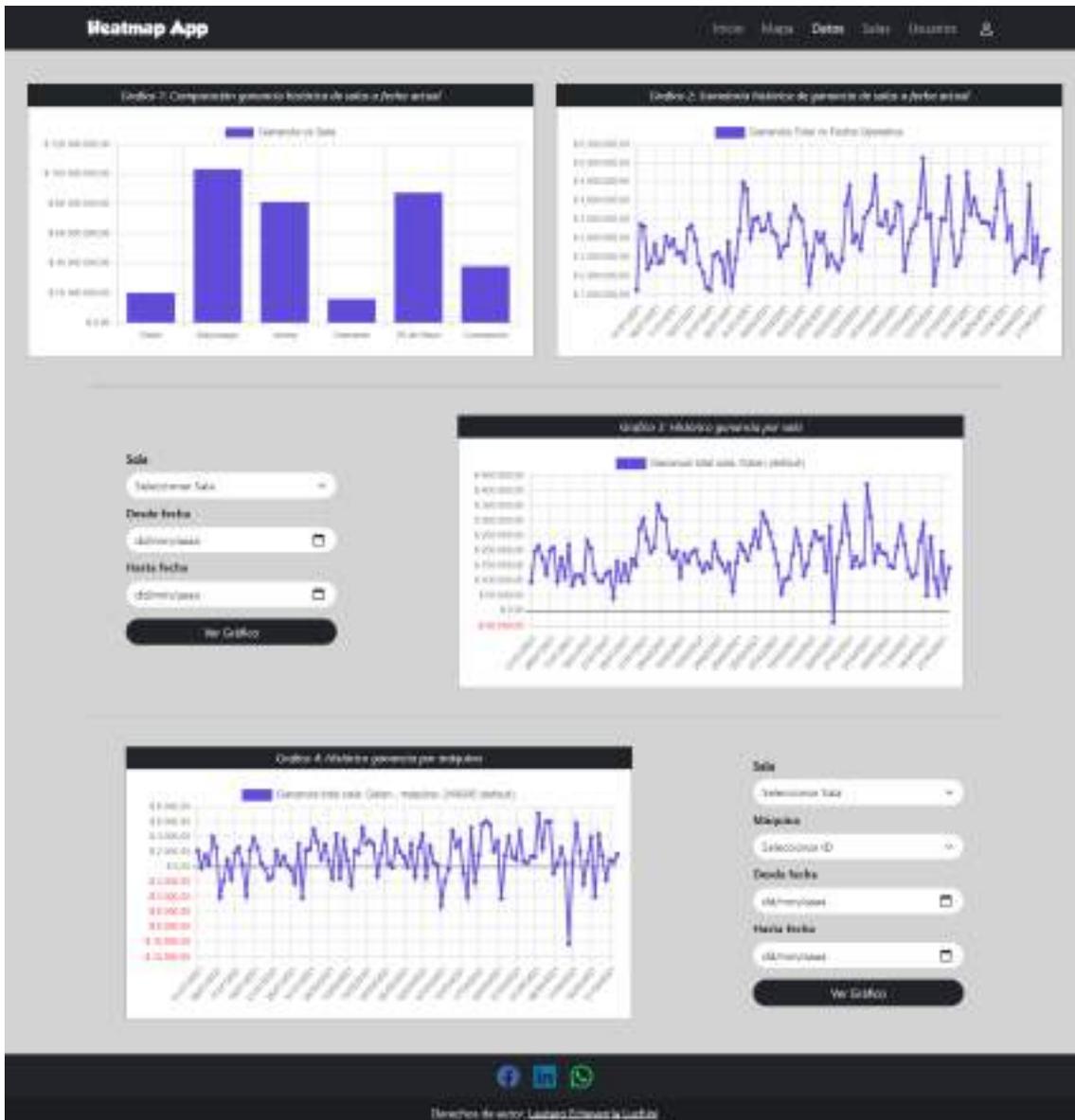
Es la interfaz de inicio o *home* del sistema donde es redirigido el usuario al ser autenticado en el Login. Esta pantalla comparte con las demás un header con el acceso a otras pantallas y un footer con las redes sociales de la empresa Neogame SA.

- Mapa:



Es la interfaz principal del sistema donde se muestra el mapa de calor de cada sala. Es posible filtrar una sala y un sector en un rango de fechas para ver una versión del mapa. También, se puede crear un mapa desde cero agregando las máquinas mediante *drag & drop* para los usuarios administradores del sistema. Las leyendas (colores) son parametrizables para ser dinámico o modificable en ciertas mediciones. En la zona inferior de la interfaz, se observa una tabla con todos los resultados correspondientes a las máquinas de la sala, estén o no en el mapa (puede existir un mapa que no tenga todas las máquinas en él). En la zona superior, se pueden ver valores estadísticos promedio del sector de la sala, generar un snapshot en otras fechas con la misma disposición del mapa (las máquinas la misma ubicación), mover las máquinas por el mapa habilitando la opción "Libre", agregar o eliminar máquinas del mapa, cambiar el criterio de consulta (win, din, dout, djp), guardar la versión o borrarla. **(Anexo III)**

- Datos:



Es la interfaz de datos estadísticos representados visualmente por gráficos lineales o de barras. Hay dos gráficos que no son parametrizables (gráfico 1 y gráfico 2) y dos gráficos que sí son parametrizables acorde a la necesidad del usuario (gráfico 3 y gráfico 4). Al igual que la interfaz Mapa, todos los datos están actualizados a la última fecha y son obtenidos por API a la base de datos. **(Anexo III)**

- Salas:

ID	Nombre	Puntaje	Ciudad	Mapa
1	Hotel Casino	10.000	Montevideo	Ver Mapa
2	Hotel 2	10.000	Montevideo	Ver Mapa
3	Hotel 3	10.000	Montevideo	Ver Mapa
4	Hotel 4	10.000	Montevideo	Ver Mapa
5	Hotel 5	10.000	Montevideo	Ver Mapa
6	Hotel 6	10.000	Montevideo	Ver Mapa
7	Hotel 7	10.000	Montevideo	Ver Mapa
8	Hotel 8	10.000	Montevideo	Ver Mapa
9	Hotel 9	10.000	Montevideo	Ver Mapa
10	Hotel 10	10.000	Montevideo	Ver Mapa
11	Hotel 11	10.000	Montevideo	Ver Mapa
12	Hotel 12	10.000	Montevideo	Ver Mapa
13	Hotel 13	10.000	Montevideo	Ver Mapa
14	Hotel 14	10.000	Montevideo	Ver Mapa
15	Hotel 15	10.000	Montevideo	Ver Mapa
16	Hotel 16	10.000	Montevideo	Ver Mapa
17	Hotel 17	10.000	Montevideo	Ver Mapa
18	Hotel 18	10.000	Montevideo	Ver Mapa
19	Hotel 19	10.000	Montevideo	Ver Mapa
20	Hotel 20	10.000	Montevideo	Ver Mapa

Es la interfaz que muestra las salas de casino de la empresa Neogame SA, con datos de ubicación geográfica de cada una. Desde esta interfaz se puede acceder a la interfaz Mapa por el botón asociado a cada fila. Los botones inhabilitados son de salas que no poseen resultados de máquinas tragamonedas en la base de datos.

- Usuarios:

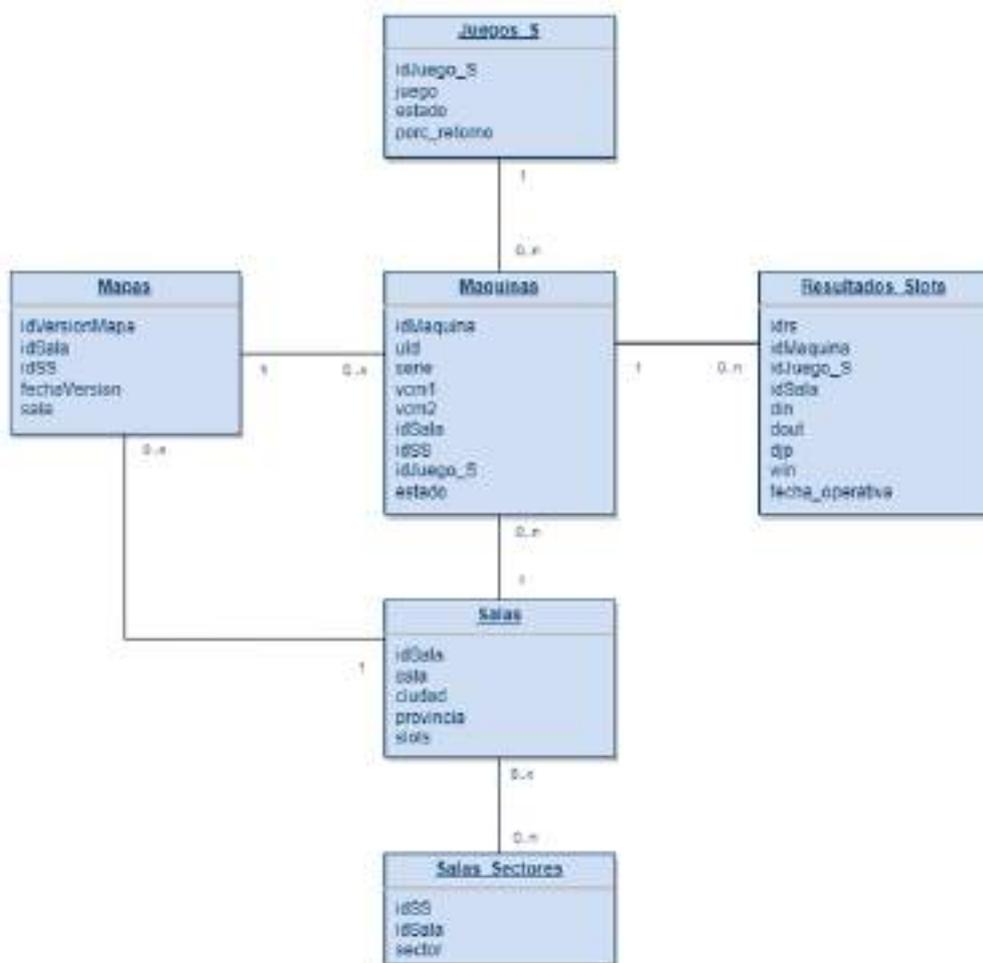
ID	Nombre	Apellido	Email	Rol
1	Admin	Admin	admin@neogame.com	1
2	Admin	Admin	admin@neogame.com	1
3	Admin	Admin	admin@neogame.com	1
4	Admin	Admin	admin@neogame.com	1
5	Admin	Admin	admin@neogame.com	1
6	Admin	Admin	admin@neogame.com	1
7	Admin	Admin	admin@neogame.com	1
8	Admin	Admin	admin@neogame.com	1
9	Admin	Admin	admin@neogame.com	1
10	Admin	Admin	admin@neogame.com	1
11	Admin	Admin	admin@neogame.com	1
12	Admin	Admin	admin@neogame.com	1
13	Admin	Admin	admin@neogame.com	1
14	Admin	Admin	admin@neogame.com	1
15	Admin	Admin	admin@neogame.com	1
16	Admin	Admin	admin@neogame.com	1
17	Admin	Admin	admin@neogame.com	1
18	Admin	Admin	admin@neogame.com	1
19	Admin	Admin	admin@neogame.com	1
20	Admin	Admin	admin@neogame.com	1

Es la interfaz de todos los usuarios del sistema. Indica datos de los usuarios y su rol (administrador u otro usuario).

6.3 Modelado de Base de Datos

Un modelo de base de datos muestra la estructura lógica de la base con sus relaciones y entidades involucradas. En el diseño de una base de datos, es una herramienta imprescindible previa a la implementación donde se definen las restricciones y limitaciones de la misma.

El siguiente modelo de diseño muestra de forma simplificada las tablas de la base de datos que fueron utilizadas en el sistema:



Las tablas mostradas en el modelo anterior se corresponden con un grupo de tablas de la base de datos MySQL utilizadas por el sistema desarrollado y el sistema ERP de forma conjunta. Si bien la base de datos contiene muchas más tablas de las mencionadas, para este proyecto y el sistema en cuestión sólo fueron necesarias estas.

La base de datos no fue creada desde cero, sino que la misma ya existía y era alimentada por datos desde el sistema ERP. Estos datos luego fueron consumidos por el sistema desarrollado conectándose a la misma.

Las tablas que ya existían en la base de datos del sistema ERP eran:

- Juegos_S
- Maquinas
- Salas
- Resultados_Slots

La representación de las columnas de las tablas en el modelo de base de datos es una simplificación como se señaló anteriormente, ya que existen más columnas en las tablas que no fueron utilizadas para el propósito de este proyecto.

Las tablas que se agregaron en la base de datos que comparten ambos sistemas son:

- Mapas
- Salas_Sectores

En la primera tabla (Mapas), se almacenan todas las versiones creadas y guardadas de los mapas por sala, sector y fechas. En la segunda tabla (Salas_Sectores), se almacenan las relaciones sala/sector permitiendo elaborar mapas para una misma sala pero diferentes sectores dentro de ella.

6.4 Seguridad

La seguridad informática en un sistema consiste en cuatro objetivos fundamentales y sus correspondientes amenazas para poder prevenirlas y utilizar distintos mecanismos de protección que lo resguarden.

Como menciona Tanenbaum (2009, p. 614), los objetivos que debe cumplir y las amenazas que puede sufrir un sistema computacional son:

Objetivo	Amenaza
Confidencialidad de los datos	Exposición de los datos
Integridad de los datos	Alteración de los datos
Disponibilidad del sistema	Negación del servicio
Exclusión de los usuarios externos	Los virus se apropian del sistema

El sistema desarrollado en este proyecto cuenta con los siguientes mecanismos de protección para la seguridad:

- Autenticación de usuarios con contraseñas hash: el proceso de autenticación para ingresar al sistema encripta la contraseña proporcionada y la convierte en un hash alfanumérico con encriptación MD5 (Message Digest Algorithm 5), un algoritmo de codificación de 128 bits que se compone de 32 caracteres hexadecimales.
- Método POST para login: el método HTTP "POST" envía al servidor una petición con los valores de usuario y contraseña que solicita acceso al sistema sin exponer dichos datos en la URL, el caché ni tampoco en el historial de navegación a diferencia del método "GET" que sí lo hace.
- Sesiones de usuario: los usuarios autenticados que ingresan al sistema tienen un sesión asociada al navegador, la cual es única por usuario y tiene un tiempo de expiración de ocho horas (tiempo de una jornada laboral). De esta manera,

si un usuario cierra el navegador o se cumplen las horas de expiración, se cerrará la sesión protegiendo al sistema del acceso de otra persona no autorizada. Además, si una persona intenta acceder a un usuario que ya está logueado, no tendrá permitido el acceso. Esta protección se implementó por medio del módulo de Node.js: `express-session`.

- Protección de cabeceras HTTP: las cabeceras retornadas por el servidor Node.js y Express.js, son protegidas a través del módulo “`helmet.js`” evitando Cross-Site-Scripting(XSS), clickjacking, entre otras amenazas.
- Control de inyección SQL: las operaciones que involucran la base de datos MySQL donde se almacenan los datos sensibles de la empresa, son realizadas todas a través de procedimientos almacenados en la base y ejecutados por API, previniendo el tipo de dato ingresado y pasados como parámetro.

Capítulo 7

Análisis de benchmarking

7.1 ¿Qué es Benchmarking?

El término “Benchmarking” surge en el año 1979 en la empresa Xerox, uno de los proveedores más grandes del mundo de tóner. En aquel entonces, la empresa estaba vendiendo productos y servicios por debajo de sus costos de producción. De esta manera, se plantea una revisión del modelo de gestión para corregir las falencias en las ventas. Por lo tanto, se hizo un estudio de la competencia y los elementos de las máquinas que vendían. Se encontró que imitando a los mismos, sería posible abaratar los costos de producción. A partir de ese año, surge el benchmarking como una herramienta importante en las industrias y sistemas productivos.

Según (Camp, 1993, p.23), “Benchmarking es la búsqueda de las prácticas de la industria que conducen a un desempeño excelente”.

En este capítulo, se hará un estudio y análisis de benchmarking de la competencia y los beneficios que ofrece el sistema desarrollado para el trabajo.

7.2 Análisis de Competencia

- Wigos

Wigos es un sistema de gestión de casinos de la empresa Winsystems que ofrece funcionalidades para administrar mesas y máquinas tragamonedas, generar reportes y estadísticas, entre otras opciones contables.



Por otro lado, permite la ejecución del sistema en *mobile* siendo altamente portable.



7.3 Diferencias y oportunidades del sistema

El sistema elaborado en este trabajo, presenta algunas ventajas respecto a los mencionados en los ejemplos anteriores. Estas ventajas son:

- Información histórica de las máquinas de casino, visto a través de mapas de calor.
- Filtros de la información según ingreso, egreso, premios grandes (jackpots) o ganancia.
- Parametrizar los valores límites superiores e inferiores para los mapas de calor dependiendo de la variable a visualizar, pudiendo mantenerlos actualizados a través del tiempo.
- Posibilidad de armar salas posicionando libremente las máquinas dentro de ellas, agregando o eliminando las mismas.
- Guardar versiones de las salas generadas para luego consultarlas.

Capítulo 8

Conclusiones Finales

8.1 Memorias del Proyecto

A modo de conclusión y realizando un análisis subjetivo de lo recorrido hasta el momento en la elaboración del proyecto, podría sintetizar que desde el punto de vista de las emociones y sentimientos que me han atravesado vivencie diferentes etapas.

En una primera etapa, se comienza una instancia de mucha incertidumbre que lleva a cierta ansiedad que aparece a partir de las dudas al no saber cómo comenzar el proyecto habiendo ya aprobado el protocolo. A diferencia de otros trabajos realizados en el marco de la Universidad, donde las herramientas eran dadas por las cátedras y se tenían ciertos seguimientos semanales, en este trabajo final de la carrera se tiene otro tipo de responsabilidad que implica un plus de trabajo de investigación. El sentimiento que genera esta libertad que se tiene al poder elegir qué hacer y cómo hacerlo lo definiría como ambiguo: por un lado es positivo pero por momentos se vuelve, en lo personal, negativo. No obstante, la incertidumbre no quedó en esos pasos, sino que se

siguió presentando en los momentos que se tenía que definir la manera en que se iba a construir el sistema y sobre cómo disponer la información que se requería. Sin embargo, gracias al esfuerzo individual, el trabajo investigativo y la recolección de experiencias de trabajos anteriores, se pudo llevar a cabo, la primera de las tareas planificadas en la confección del sistema.

En una segunda etapa, habiendo concretado el análisis y diseño, se da lugar al momento del desarrollo y testeo del sistema. En contraste a la primera etapa, no hubo tantas dudas y se desarrolló de manera constante el trabajo. Si bien hubo ciertos altos y bajos, como en todo proceso de elaboración, los encuentros y las correcciones del director Fernando Genin hicieron al proyecto más ameno, permitiendo enriquecerlo en cada paso.

En una tercera etapa, la última del trabajo, se comenzó a escribir el presente documento final, el cual plasma todo lo trabajado en estos largos meses. Esta fue la más difícil de las tres etapas, ya que fue un sistema desarrollado, en primer lugar de manera individual, y en segundo lugar, uno de los más importantes y extensos en la carrera de Ingeniería Informática hasta el momento. Es por ello, que al no tener la suficiente formación o experiencia en torno a la escritura de este tipo de documentos -ya que la carrera es muy práctica- no fue fácil llevar a cabo la redacción de los conceptos aplicados así como tampoco cumplir con las normas que se debía tener para poder presentar un trabajo final.

Desde el primer momento, el proceso de redacción y escritura del trabajo fue desafiante. La tarea se volvía difícil a la hora de sentarse a plasmar en palabras todo el trabajo efectuado en la práctica ya que surgían diferentes dudas. Las principales tuvieron que ver con los tiempos verbales y las personas gramaticales a utilizar (confusión respecto a redactar en primera o tercera persona). En segundo lugar, se generaba confusión a la hora de explicar en la profundidad de detalle de los conceptos, es decir que, me encontraba en la línea delgada entre escribir detalles innecesarios y/o escasos o escribir por demás. Entre otras, esta complejidad se sumó

en lo personal para finalizar el proyecto. De todas maneras al final, se logró desde mis expectativas, estar conforme con el documento elaborado para este Proyecto Final.

8.2 Cumplimiento y evolución de los objetivos

En cuanto al objetivo general y los objetivos específicos definidos al comienzo del proyecto, se podría decir que se cumplieron en su totalidad sin excepciones. Se logró construir una herramienta tecnológica de calidad para la empresa que aporte valor y genere una transformación digital en las operaciones diarias que utilicen los distintos empleados. El proyecto aportó a la empresa una nueva forma de trabajar en la disposición de las máquinas, atendiendo mediciones específicas, aprovechando mayores beneficios y generando menores pérdidas económicas. **(Anexo V)**

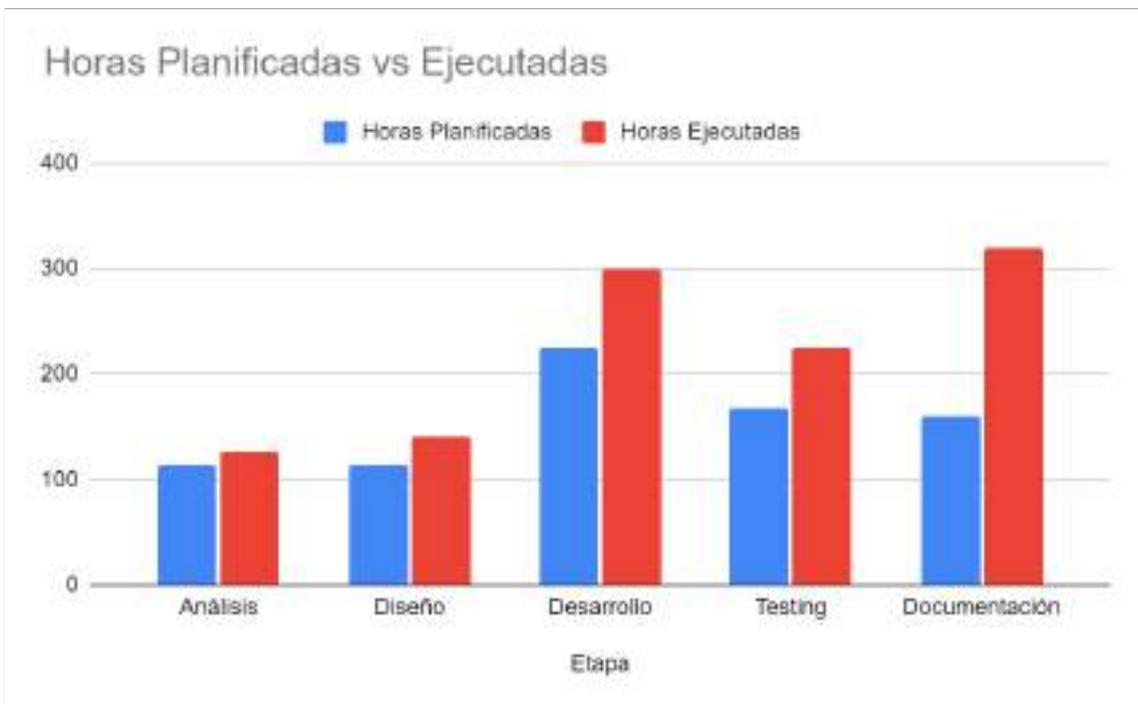
La evolución de los objetivos durante el proyecto fue de forma secuencial y ordenada. Cada objetivo, se trabajó de inicio a fin, buscando una resolución completa de cada uno de ellos para poder continuar con el siguiente de manera prolija y metódica. Por otra parte, los objetivos fueron evolucionando con el tiempo -aunque su base fundamental se mantuvo- y gracias a las reuniones con la empresa, las cuales aportaban ideas y nutrían el proyecto para nuevas modificaciones que lo reestructuraban.

8.3 Planificación vs Ejecución

Este apartado pretende plasmar la contraposición que encuentro entre aquello que fue planificado y las dificultades que se pusieron en juego a la hora de llevar a cabo la ejecución. En primer lugar, el tiempo total del proyecto desarrollado no coincidió con los tiempos planificados inicialmente. La estimación de plazos del proyecto presentado en el cronograma del protocolo era de 780 horas de trabajo, sin embargo la duración se extendió a 1110 horas, dando una diferencia notoria de 330 horas de más, lo que representa aproximadamente casi un 50% de demora en la finalización del proyecto. Si

bien por distintos factores la estimación del tiempo no fue acertada, el desarrollo se llevó a cabo con los requisitos necesarios para una eficaz ejecución evaluando la necesidad de más tiempo de trabajo.

El siguiente gráfico muestra una comparación entre las horas estimadas y las horas ejecutadas reales:



Etapa	Horas Planificadas	Horas Ejecutadas
Análisis	114	126
Diseño	114	140
Desarrollo	224	300
Testing	168	224
Documentación	160	320
Total Horas	780	1110

Por otro lado, en cuanto a los lenguajes de programación planificados a utilizar me veía capacitado en todos los mencionados a lo largo del proyecto. Pese a ello, al momento de ejecutarlos en el desarrollo del sistema tuve que formarme, investigar, llevar a cabo lectura de libros e interiorizarme con algunos conceptos que no conocía hasta el momento.

8.3.1 Desvíos

Debido a una serie de factores inesperados que fueron surgiendo a medida que se trabajaba dentro del proyecto, la duración terminó siendo mayor a la programada al comienzo del trabajo. Los desvíos detectados fueron:

- La elaboración de la documentación llevó el doble del tiempo previsto, dado que fue más laborioso de lo esperado, no se tenían los recursos pensados para realizarla, numerosas dudas en la redacción, entre otras cosas.
- Etapas de desarrollo y testing mayores a las previstas, ya que se construyeron componentes y funcionalidades que no fueron tenidas en cuenta en un principio.
- Etapas de análisis y diseño levemente mayores a las previstas, por surgir modificaciones de requerimientos funcionales luego de haber completado gran parte del desarrollo del sistema.
- Formación e investigación en torno a los lenguajes utilizados.
- Desmotivación personal durante los altos y bajos de la elaboración y el no poder cumplir con los tiempos propuestos.
- Más horas de trabajo, por ende, menos horas disponibles para dedicarle al proyecto.

8.4 Obstáculos del proyecto

Durante el desarrollo, no se presentaron grandes obstáculos externos que impidan su realización. El principal obstáculo identificado previo a comenzar, fue la pandemia

aunque esta no generó inconvenientes para llevar a cabo el proyecto, sino que benefició el entorno digital. Gracias a la virtualidad, las reuniones y seguimientos se hacían por medio de videollamadas, donde el sistema se podía presentar y mostrar en su totalidad, sumando miradas y opiniones críticas que hicieron a su desarrollo y crecimiento. Por último, la información estuvo siempre a disposición para poder avanzar con el trabajo.

8.5 Aprendizaje

El aprendizaje obtenido desarrollando el proyecto fue muy enriquecedor, dejando una amplia cantidad de herramientas nuevas para desempeñarme dentro del ámbito de la Ingeniería Informática. Al mismo tiempo, fue importante adquirir dicho aprendizaje de forma autodidacta, forjando una actitud proactiva y de constantes desafíos que se suman al *expertise* personal. Asimismo, se aprendió sobre el negocio de los casinos y su funcionamiento, incorporando muchos conceptos y procesos del rubro en cuestión.

Los puntos aprendidos más importantes fueron:

- Estimar los proyectos con plazos pesimistas para no retrasar las entregas por no poder cumplir con lo planificado.
- El modelo en cascada es menos eficiente que las metodologías ágiles.
- Utilizar frameworks que faciliten el desarrollo web frontend a pesar de tener que invertir tiempo en formación, ya que a la larga es menos costoso en horas totales (formación y desarrollo).
- La documentación de un sistema es larga y demanda muchas horas de trabajo. Por ende, tenerlo en cuenta en las consideraciones previas del proyecto.

Bibliografía

1. Arboleda, L. (2004) Servicios web: distribución e integración." *Sistemas & Telemática*, 2(4), 107-120.
2. Baca Urbina, G. (2001) *Evaluación de proyectos (Cuarta edición)*. México DF: McGraw-Hill.
3. Booch, G., Jacobson, I. y Rumbaugh, J. (2000) *El proceso unificado de desarrollo de software*. Madrid: Pearson Educación.
4. Brown, E. (2019) *Web Development with Node and Express: Leveraging the JavaScript Stack*. O'Reilly Media.
5. Camp, R. (1993) *Benchmarking*. México: Editorial Panorama Editorial S.A.
6. Coutiño, L. (2012) *Análisis de sistemas de información*. México: Red Tercer Milenio.
7. Díaz, J, Harari, I. y Amadeo, A. (2013) *Guía de recomendaciones para diseño de software centrado en el usuario*. La Plata: EDULP.
8. Elmasri R. y Navathe S. B. (1989) *Fundamentos de Sistemas de Bases de Datos. (Quinta edición)*. Addison-Wesley.
9. Flanagan, D. (2020) *JavaScript - The Definitive Guide*. O'Reilly Media.
10. Fuentes, M., Ojeda, C. y González, P. (2019) *Fundamentos de Ingeniería de Software*. México DF: Universidad Autónoma Metropolitana.
11. Gamma, E., Helm, R., Johnson, R. y Vlissides, J. (2003) *Patrones de diseño. Elementos de software orientado a objetos reutilizables*. Addison-Wesley.
12. Hall, E. (1998) *Managing Risk: Methods for Software Systems Development (1st Edition)*. Addison-Wesley.
13. Herron, D. (2020) *Node.js Web Development: Server-side web development made easy with Node 14 using practical examples (5th Edition)*. Birmingham: Packt Publishing Limited.
14. Huerta, D. (2020) *Análisis FODA o DAFO*. Madrid: Bubok Publishing.
15. Kanat-Alexander, M. (2012) *Code Simplicity: The Fundamentals Of Software*. O'Reilly Media.

16. Kendall, J. y Kendall, K. (2011) *Análisis y Diseño de Sistema (Octava edición)*. Prentice Hall.
17. Larman, C. (2003) *UML y Patrones. Una introducción al análisis y diseño orientado a los objetos y al proceso unificado*. Madrid: Pearson Educación.
18. Lerma-Blasco, R., Murcia, J. y Mifsud, E. (2013) *Aplicaciones web*. McGraw-Hill.
19. Lorenzon, E. (2020) *Sistemas y organizaciones. Parte I: Teoría General de Sistemas Aplicada. Parte II: Las organizaciones. Su funcionamiento como sistema*. La Plata: EDULP.
20. Marini, E. (2012) "El Modelo Cliente/Servidor". [Online]. Consultado el 7 de diciembre de 2021 en <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>
21. Martin, R. (2018) *Clean Architecture. A craftsman's guide to software structure and design*. Pearson Education.
22. Mateu, C. (2004) *Desarrollo de aplicaciones web*. Barcelona: Eureka Media.
23. Montealegre Tovar, Mauricio (2008). *Definición de proyecto*. Bogotá: Servicio Nacional de Aprendizaje Centro de Servicios Administrativos Regional Distrito Capital.
24. Montilva, J. (1999) *Desarrollo de Sistemas de Información*. Mérida: Universidad de los Andes.
25. Plaza, S., Ramirez, N. y Acosta, C. (2015) *API de servicios web orientados a accesibilidad*. Universidad Complutense de Madrid
26. Pressman, R. (2010) *Ingeniería del software. Un enfoque práctico (Séptima edición)*. Nueva York: McGraw-Hill.
27. Reese, G. (2007) *MySQL Pocket Reference 2e (Segunda edición)*. O'Reilly Media.
28. Rodríguez, R. y Cobas, M. (s.f) "Metodología de evaluación de impactos de proyectos de investigación." [Online]. Consultado el 15 de enero de 2022 en https://inis.iaea.org/collection/NCLCollectionStore/_Public/45/078/45078486.pdf
29. Rubio-Moraga, Á. (2003) *Internet y enseñanza: la educación virtual*. España: Edipo.
30. Sapag Chain, N. y Sapag Chain, R. (2008) *Preparación y evaluación de proyectos (Quinta edición)*. Bogotá: McGraw-Hill.
31. Sommerville, I. (2011) *Ingeniería del software*. Addison-Wesley.

32. Tanenbaum, A. (2009) *Sistemas operativos modernos (Tercera edición)*. México: Pearson Educación.
33. Thompson, A. y Strikland, K.F.C. (1998) *Dirección y administración estratégicas. Conceptos, casos y lecturas*. México: McGraw-Hill Interamericana.
34. Trejo, E., Trejo, N. y Zúñiga, J. (2016) "Análisis FODA del sector lácteo: un estudio de caso." *Revista de Planeación y Control Microfinanciero*, 2(4), 8-22.
35. Vanzetti, J. (2006) *Un modelo del proceso de desarrollo de software guiado por la traceability*. La Plata: UNLP.
36. Yourdon, E. (1993) *Análisis estructurado moderno*. México: Prentice-hall Hispanoamérica.

Anexo I: Lenguajes y herramientas del sistema

HTML y CSS

El Lenguaje de Marcas de Hipertexto (HTML) es un componente básico para cualquier sistema web. Este define el significado y la estructura del contenido. Otra característica importante es que “HTML usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas HTML son palabras clave y atributos rodeados de los signos mayor y menor.” (Gauchat, 2012, p. 3). La palabra “Hipertexto” hace referencia a los enlaces que conectan varias páginas web entre sí. A través de estos, el usuario podrá navegar entre páginas que son distintos documentos HTML. Actualmente, la versión utilizada es HTML5 la cual combina estructura, estilos (CSS) y funcionalidades (JS).

Las Hojas de Estilo en Cascada (CSS) son un elemento fundamental en el diseño web. Estos archivos se asocian con los documentos HTML para trabajar en forma conjunta. Si bien, HTML define una estructura del sistema, por otro lado, CSS define los estilos vinculados a esa estructura. Por lo tanto, un sistema web sin estilos, es decir, HTML puro, podrá ser utilizado pero no tendrá los detalles estéticos que pueden proveer las hojas de estilo CSS. La última versión de CSS es CSS3 y es la usada por HTML5.



Bootstrap

Bootstrap es un framework CSS desarrollado por Twitter en el año 2010 con el objetivo de estandarizar las herramientas de la compañía. Una de las funciones más importantes de Bootstrap es el diseño web responsive, definido por González & Marcos (2013, p.4) como “una serie de técnicas y pautas de diseño que permiten adaptar sitios web al entorno de navegación del usuario, entendiendo como entorno de navegación la multiplicidad de dispositivos, móviles o no, por medio de los cuales los usuarios pueden acceder y navegar en internet.”



Por otra parte, Bootstrap proporciona una biblioteca de componentes que facilitan el desarrollo de un sistema web, ya que tienen estructuras y estilos predefinidos. La última versión de Bootstrap es la v5.1.

JavaScript

El lenguaje de programación JavaScript es un lenguaje que se emplea en los sistemas web, tanto en el frontend interactuando con el cliente (páginas HTML) o en el backend funcionando con el servidor (Node.js, Apache CouchDB y Adobe Acrobat). Es un lenguaje interpretado, no tipado, basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativo y declarativo.

En el frontend, existen diversos frameworks basados en JS que son utilizados por desarrolladores para aplicaciones interactivas y escalables. Estos cuentan con

numerosas herramientas para aplicar en las interfaces de usuarios (UI). Los tres más conocidos son: Angular, React y Vue.js.



Si bien la utilización de un framework simplifica la tarea de desarrollo, en ciertas ocasiones no es necesario recurrir a uno debido al tipo y tamaño del proyecto. Un framework hace que los archivos del sistema sean más pesados y, por tal motivo, afecten directamente en el rendimiento.

Este proyecto no utilizó ninguno de los frameworks mencionados en su desarrollo.

JQuery

JQuery es un framework de JavaScript, de código abierto, muy utilizado en las aplicaciones web por sus herramientas de interacción con los documentos HTML.

Como lo mencionan algunos autores:

Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cosa muy interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc.

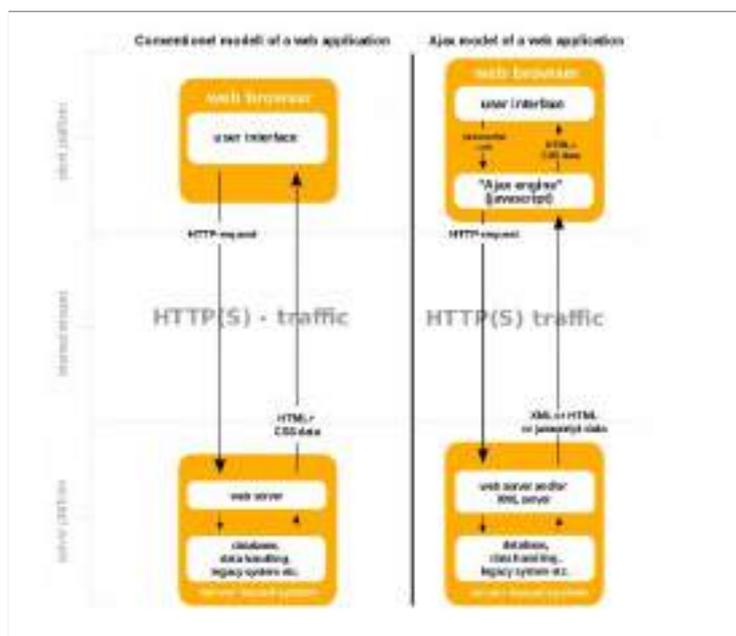
(Alvarez , 2010, p.3)

La última versión estable es la 3.5.0 lanzada el 20 de abril de 2020 y fue la utilizada en el sistema del trabajo.

AJAX

AJAX (Asynchronous JavaScript And XML) es una técnica de desarrollo que combina HTML, CSS, JS, DOM (Document Object Model), XML (eXtensible Markup Language) y un objeto XMLHttpRequest. Mediante estas tecnologías, permite a un sistema web trabajar de forma asíncrona, realizando una o más peticiones al servidor, y actualizar el sistema cuando la respuesta del servidor es efectuada. De este modo, no es necesario refrescar el sistema o la página cuando la respuesta es recibida. Las principales ventajas de AJAX son: mejor experiencia de usuario (UX), optimización de recursos (velocidad), mayor portabilidad y usabilidad.

En el gráfico, se observa el funcionamiento de dos aplicaciones web, en las cuales una no utiliza AJAX mientras que la otra sí. La diferencia fundamental, radica en que la segunda aplicación, hará una petición al navegador que activará una XMLHttpRequest. Esta se ejecutará en segundo plano mostrando los datos en la interfaz una vez que sean recibidos sin recargar la página.



ChartJS

La biblioteca ChartJS es de JavaScript de código abierto que incluye distintos tipos de gráficos para la representación de datos. Los gráficos que admite son: línea, barra, radar, torta, polar, burbuja, dispersión y área. ChartJS es una de las librerías de gráficos de JS más votadas y elegidas entre los desarrolladores.

NodeJS

NodeJS es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. NodeJS utiliza el lenguaje JavaScript en la capa del servidor y no en la del cliente. Este entorno fue ideado en 2009 y orientado a eventos asíncronos para construir aplicaciones escalables.



A través de NodeJS, es posible crear un servidor en un host y puerto específico. Luego ese servidor se pondrá a escuchar peticiones entrantes de clientes que deseen establecer una conexión con él.

Express

Express.js o Express es un framework de NodeJS mínimo, flexible y de código abierto. Además, es muy popular por las funciones que ofrece para el desarrollo de aplicaciones web y APIs. Express está basado en NodeJS y depende de este para su ejecución.

Los mecanismos ofrecidos por Express posibilitan:

- Renderizar dinámicamente vistas HTML pasando argumentos a una plantilla, como por ejemplo EJS, HBS o PUG.
- Configurar el servidor de NodeJS en un puerto específico.
- Añadir una capa middleware para responder peticiones HTTP.

Paquetes de NodeJS

En NodeJS existen un enorme cantidad de paquetes disponibles para usar en el desarrollo. Los paquetes son un grupo de archivos .js que tienen código reutilizable y una función específica asociada. Brindan una alta facilidad y ahorro a la hora de desarrollar una aplicación.

En este trabajo los paquetes utilizados fueron:

- CORS (Cross Origin Resource Sharing): restringe o concede acceso a una petición HTTP a un recurso de un servidor analizando dónde se originó la misma.
- Dotenv: permite cargar variables de entorno usando un archivo .env independiente de los archivos del sistema.
- EJS (Embedded JavaScript): genera una estructura HTML con JavaScript plano utilizando plantillas que son renderizadas.
- Express-Session: crea un sesión middleware para acceder al servidor.
- Helmet: es un middleware que funciona entre Node.js y Express.js cuya principal función es proteger las cabeceras HTTP retornadas del servidor.
- MD5: aplica un algoritmo de hashing a un string pasado como parámetro. Para el trabajo se utilizó encriptando una clave de usuario y comparando con la clave encriptada en la base de datos MySQL.
- MySQL: utilizado para conexiones y consultas con bases de datos MySQL.

- Nodemon: reinicia el servidor de NodeJS cuando se realiza un cambio y se guarda el archivo modificado.

MySQL

MySQL se puede definir como “un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento.” (Casillas Santillán, Ginestá y Pérez Mora, 2014, p. 5). Así mismo, es un SGBD de licencia gratuita y de bases de datos relacionales. La gestión de las bases de datos de MySQL es por medio de distintas interfaces de usuario. Las más conocidas son: MySQL Workbench, DBeaver, phpMyAdmin, entre otras. Para el sistema de este trabajo, se utilizó phpMyAdmin la cual corre en un navegador web.

XAMPP

XAMPP es una herramienta de desarrollo y libre distribución de Apache, utilizada para aplicaciones web, ya que ofrece la posibilidad de emular un servidor web Apache y brindar acceso a bases de datos MySQL o MariaDB. El nombre es un acrónimo de Linux, Apache, MySQL/MariaDB, PHP y Perl. En este trabajo, sólo se hizo uso del servidor Apache y la base de datos MySQL.

Anexo II: Casos de Uso

Los casos de uso del sistema desarrollado en el proyecto se detallan a continuación:

Caso de Uso	#01. Login de usuario	
Actores	Gerente, Encargado, Empleado	
Propósito	Ingresar al sistema para poder utilizarlo.	
Descripción	El usuario Gerente/Encargado/Empleado desea loguearse al sistema para acceder al mismo y utilizar sus funcionalidades.	
Precondiciones	Poseer un usuario ya registrado en el sistema ERP.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página web del sistema.
	2	El sistema devuelve la página de Login con un formulario.
	3	El usuario completa los datos con nombre de usuario, contraseña e intenta ingresar.
4	El sistema verifica los datos ingresados y permite el acceso del usuario a la página principal del mismo.	
Cursos Alternos	Paso	Evento
	1	El usuario no puede acceder a la página web, ya que no tiene conexión a Internet.
4	El sistema no permite el ingreso del usuario al sistema, ya que los datos no corresponden con la validación del mismo.	

Caso de Uso	#02. Visualizar Salas del sistema	
Actores	Gerente, Encargado, Empleado	
Propósito	Visualizar las salas de casinos de la empresa.	
Descripción	El usuario Gerente/Encargado/Empleado desea ver las salas de casinos de la empresa, las cuales están vinculadas con el sistema ERP de la misma.	
Precondiciones	Salas existentes en la base de datos del sistema.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del

		sistema.
	2	El usuario selecciona la pestaña Salas dentro del sistema.
	3	El sistema devuelve la página Salas, donde éstas están organizadas en una tabla según su localidad y nombre correspondiente.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Salas, ya que perdió su conexión a Internet al seleccionar dicha pestaña.

Caso de Uso	#03. Visualizar Usuarios del sistema	
Actores	Gerente, Encargado, Empleado	
Propósito	Visualizar todos los usuarios del sistema.	
Descripción	El usuario Gerente/Encargado/Empleado desea ver todos los usuarios del sistema.	
Precondiciones	Usuarios existentes en la base de datos del sistema.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Usuarios dentro del sistema.
	3	El sistema devuelve la página Usuarios, donde éstos están organizadas en una tabla según su ID, nombre y apellido correspondiente.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Usuarios, ya que perdió su conexión a Internet al seleccionar dicha pestaña.

Caso de Uso	#04. Crear Mapa de Sala	
Actores	Gerente, Encargado	
Propósito	Crear un mapa de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado desea generar una nueva versión de un mapa de un sector en una sala en un rango de fechas específico.	
Precondiciones	Sector de sala seleccionado contenga máquinas en las	

	fechas indicadas.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario selecciona el botón Crear Mapa.
	5	El sistema lanza un popup con los campos requeridos para crear una nueva versión de un mapa.
	6	El usuario completa los campos: Sala, Sector, Cantidad de Columnas (opcional), Fecha Versión, Fecha Inicial y Fecha Final.
	7	El sistema valida los datos ingresados y genera un mapa vacío de la sala y sector correspondientes.
8	El usuario puede agregar una o más máquinas y luego guardar la versión del mapa.	
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	7	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.

Caso de Uso	#05. Ver Mapa de Sala	
Actores	Gerente, Encargado, Empleado	
Propósito	Visualizar un mapa existe de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado/Empleado desea ver un mapa de una sala existente en un rango de fechas determinado.	
Precondiciones	Existe un mapa de una sala en las fechas indicadas.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro

		del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	6	No existen versiones de mapas en los rangos elegidos por el usuario.

Caso de Uso	#06. Agregar Máquina a Mapa de Sala	
Actores	Gerente, Encargado	
Propósito	Agregar una máquina a un mapa de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado desea agregar una máquina a un mapa existente de una sala y sector asociados.	
Precondiciones	Mapa de sala creado y guardado en la base de datos. Máquina existente en el sector.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.

	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
	9	El usuario escoge la máquina a agregar al mapa.
	10	El sistema posiciona la máquina en el mapa.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	8	La máquina no se encuentra disponible para agregar, ya que fue retirada del sector.

Caso de Uso	#07. Eliminar Máquina de Mapa de Sala	
Actores	Gerente, Encargado	
Propósito	Eliminar una máquina de un mapa de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado desea eliminar una máquina de un mapa existente de una sala y sector asociados.	
Precondiciones	Mapa de sala creado y guardado en la base de datos. Máquina existente en el sector y posicionada en el mapa.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de	

		fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
	9	El usuario escoge la máquina a eliminar del mapa.
	10	El sistema borra la máquina del mapa y la agrega al dropdown de máquinas disponibles para agregar.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	8.1	La máquina no se encuentra disponible para eliminar, ya que fue retirada del sector.
	8.2	La máquina no se encuentra disponible para eliminar, ya que fue eliminada por otro usuario y está disponible en el dropdown de máquinas para agregar.

Caso de Uso	#08. Editar Mapa de Sala	
Actores	Gerente, Encargado	
Propósito	Editar un mapa de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado desea editar las máquinas de un mapa existente de una sala y sector asociados. Para ello, utiliza un método <i>Drag & Drop</i> .	
Precondiciones	Mapa de sala creado y guardado en la base de datos. Máquinas agregadas al mapa.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.

	5	El usuario selecciona el botón Ver Mapa.
	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
	9	El usuario escoge las máquinas a editar y las mueve dentro del mapa.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.

Caso de Uso	#09. Guardar Versión de Mapa de Sala	
Actores	Gerente, Encargado	
Propósito	Guardar una versión de un mapa de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado desea guardar la versión de un mapa existente de una sala y sector asociados.	
Precondiciones	Mapa de sala creado y guardado.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
8	El sistema devuelve el mapa de la versión escogida por el usuario.	

	9	El usuario edita el mapa, generando una nueva modificación para guardar.
	10	El usuario selecciona botón Guardar Versión.
	11	El sistema lanza un popup de confirmación y luego guarda la versión modificada en la base de datos.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	11	El usuario se arrepiente y cancela el proceso.

Caso de Uso	#10. Borrar Versión de Mapa de Sala	
Actores	Gerente, Encargado	
Propósito	Borrar una versión de un mapa de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado desea guardar la versión de un mapa existente de una sala y sector asociados.	
Precondiciones	Mapa de sala creado y guardado.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
9	El usuario selecciona el botón Borrar Versión.	

	10	El sistema lanza un popup de confirmación y luego elimina la versión de la base de datos.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	10	El usuario se arrepiente y cancela el proceso.

Caso de Uso	#11. Ver Datos Estadísticos de Mapa de Sala	
Actores	Gerente, Encargado, Empleado	
Propósito	Ver datos estadísticos de un mapa de una sala en un rango de fechas.	
Descripción	El usuario Gerente/Encargado/Empleado desea observar los datos estadísticos de ganancia, ingreso, egreso y jackpot promedio de un mapa de una sala en el rango de fechas asociado a la versión del mapa.	
Precondiciones	Existe un mapa de una sala en las fechas indicadas.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
	9	El usuario selecciona el botón "%" de datos estadísticos.
10	El sistema lanza un popup con la información: Fecha Inicial, Fecha Final, valores promedio	

		ganancia, ingreso, egreso y jackpot entre fechas.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.

Caso de Uso	#12. Hacer Nuevo Snapshot de Mapa de Sala	
Actores	Gerente, Encargado, Empleado	
Propósito	Ver una captura instantánea de un mapa de una sala en una fecha o un rango de fechas especificadas.	
Descripción	El usuario Gerente/Encargado/Empleado desea observar los datos estadísticos de ganancia, ingreso, egreso y jackpot promedio de un mapa de una sala en el rango de fechas asociado a la versión del mapa.	
Precondiciones	Existe un mapa de una sala en las fechas indicadas.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
	9	El usuario selecciona el botón "📷" de Nuevo Snapshot.
10	El sistema lanza un popup con los campos a completar de Fecha Inicial y Fecha Final de Nuevo Snapshot.	

	11	El usuario completa Fecha Inicial y Fecha Final.
	12	El sistema valida fechas y devuelve mapa modificado en parámetros de visualización de color, con los valores asociados a las nuevas fechas ingresadas.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	12.1	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	12.2	El sistema lanza un Alert de error indicando al usuario que las fechas ingresadas son erróneas.

Caso de Uso	#13. Ver Mapa de Sala con Diferentes Parámetros de Visualización	
Actores	Gerente, Encargado, Empleado	
Propósito	Ver un mapa de una sala en un rango de fechas con diferentes parámetros de visualización.	
Descripción	El usuario Gerente/Encargado/Empleado desea observar el mapa de una sala en un rango de fechas por otro parámetro de visualización, entre ellos: ganancia, ingreso, egreso y jackpot promedio.	
Precondiciones	Existe un mapa de una sala en las fechas indicadas.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
	6	El sistema lanza un popup con las versiones de mapas creadas para la sala en el rango de fechas especificadas.

	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
	9	El usuario selecciona el botón Win.
	10	El sistema despliega un dropdown con los parámetros de ingreso permitidos para ver el mapa.
	11	El usuario selecciona el parámetro deseado.
	12	El sistema devuelve el mapa con los valores del nuevo parámetro de ingreso, modificando los colores de las máquinas.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.

Caso de Uso	#14. Modificar Parámetros de Leyendas de Mapa de Sala	
Actores	Gerente, Encargado, Empleado	
Propósito	Ver un mapa de una sala en un rango de fechas con diferentes valores promedio del parámetro de visualización.	
Descripción	El usuario Gerente/Encargado/Empleado desea observar el mapa de una sala en un rango de fechas por el mismo parámetro de visualización pero con otro valor promedio de dinero.	
Precondiciones	Existe un mapa de una sala en las fechas indicadas.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Mapa dentro del sistema.
	3	El sistema devuelve la página de Mapa.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Mapa.
6	El sistema lanza un popup con las versiones de	

		mapas creadas para la sala en el rango de fechas especificadas.
	7	El usuario escoge la versión requerida.
	8	El sistema devuelve el mapa de la versión escogida por el usuario.
	9	El usuario selecciona el botón Modificar en la sección Leyendas.
	10	El sistema lanza un popup con los campos a completar de Valor Promedio Inferior (color rojo) y Valor Promedio Superior (color verde).
	11	El usuario completa los valores promedio inferior y superior.
	12	El sistema valida valores ingresados y devuelve mapa modificado en parámetros de visualización de color, con los nuevos valores ingresados en el mismo rango de fechas en el cual se encontraba el mapa.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	12.1	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	12.2	El sistema lanza un Alert de error indicando al usuario que las fechas ingresadas son erróneas.

Caso de Uso	#15. Visualizar Gráficos Estadísticos 1 y 2: Ganancia Por Sala y Ganancia Por Día Acumulada
Actores	Gerente, Encargado, Empleado
Propósito	Ver los gráficos estadísticos 1 y 2 en la pestaña Datos
Descripción	El usuario Gerente/Encargado/Empleado desea observar los gráficos 1 y 2 de la pestaña Datos, asociados a la ganancia acumulada por sala en gráfico de barras (1) y a la ganancia sumada de todas las salas acumulada por día en gráfico lineal (2).
Precondiciones	Existen datos en la base de datos de movimientos de dinero en las salas.

Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Datos dentro del sistema.
	3	El sistema devuelve la página de Datos, donde los gráficos 1 y 2 se encuentran en la zona superior de la página.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Salas, ya que perdió su conexión a Internet al seleccionar dicha pestaña.

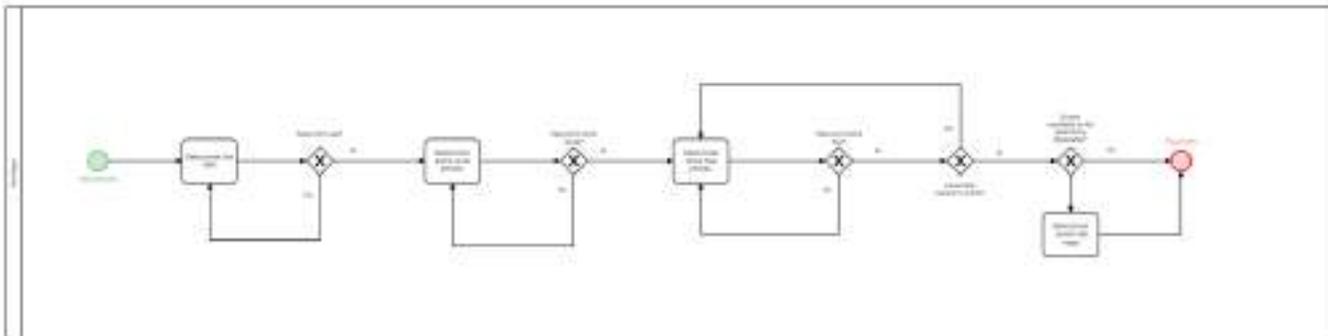
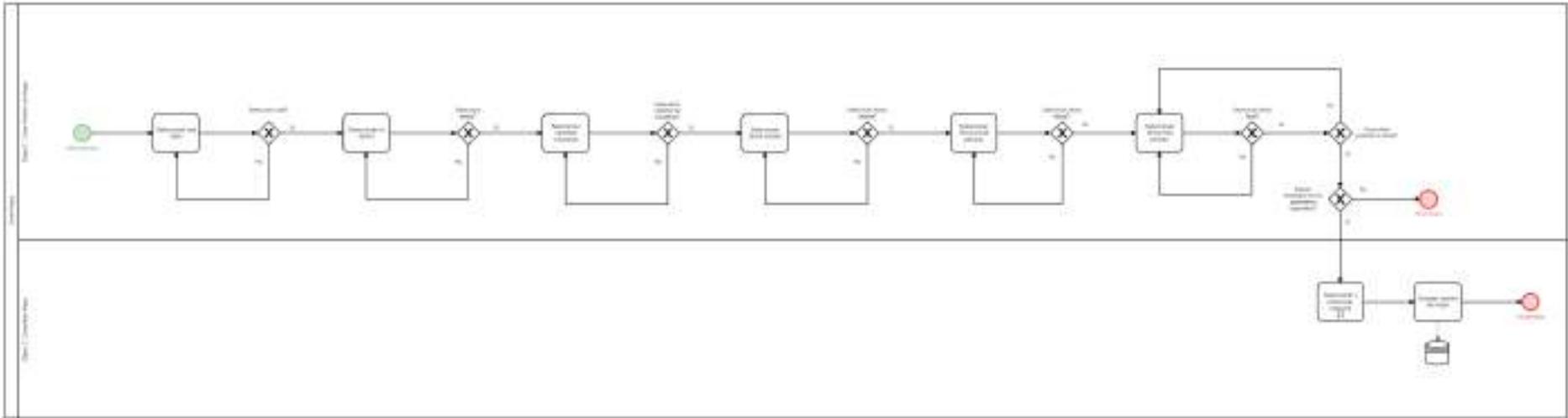
Caso de Uso	#16. Visualizar Gráfico Estadístico 3: Ganancia Por Sala en Fechas	
Actores	Gerente, Encargado, Empleado	
Propósito	Ver el gráfico estadístico 3 en la pestaña Datos	
Descripción	El usuario Gerente/Encargado/Empleado desea observar el gráfico 3 de la pestaña Datos, asociado a la ganancia por día de una sala en gráfico lineal en un rango de fechas especificadas.	
Precondiciones	Existen datos en la base de datos de movimientos de dinero en la sala.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Datos dentro del sistema.
	3	El sistema devuelve la página de Datos, donde el gráfico 3 se encuentra en la zona media de la página.
	4	El usuario completa Sala, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Gráfico.
	6	El sistema valida datos ingresados y muestra gráfico de ganancia por día desde Fecha Inicial hasta Fecha Final de la sala seleccionada.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al

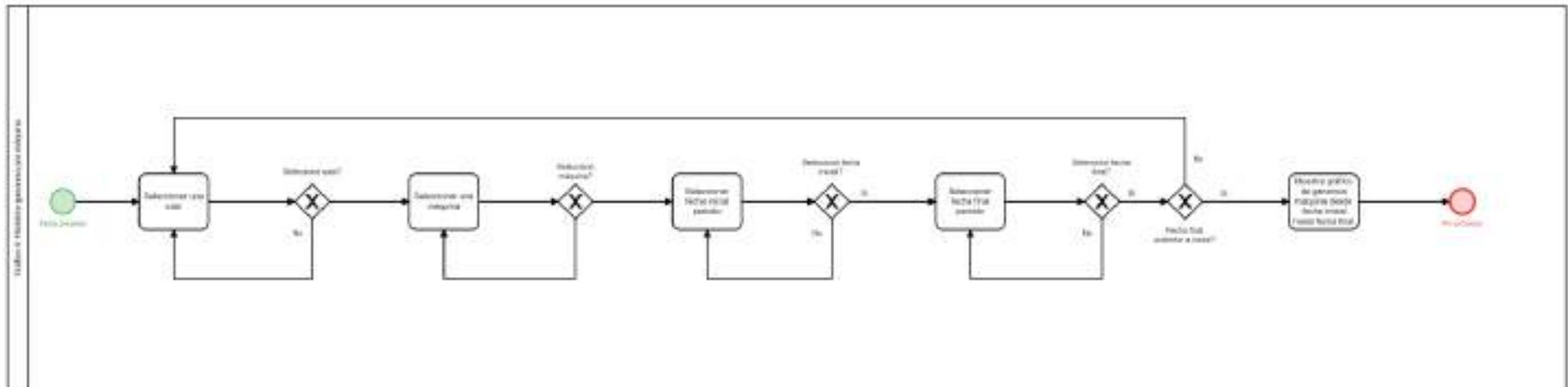
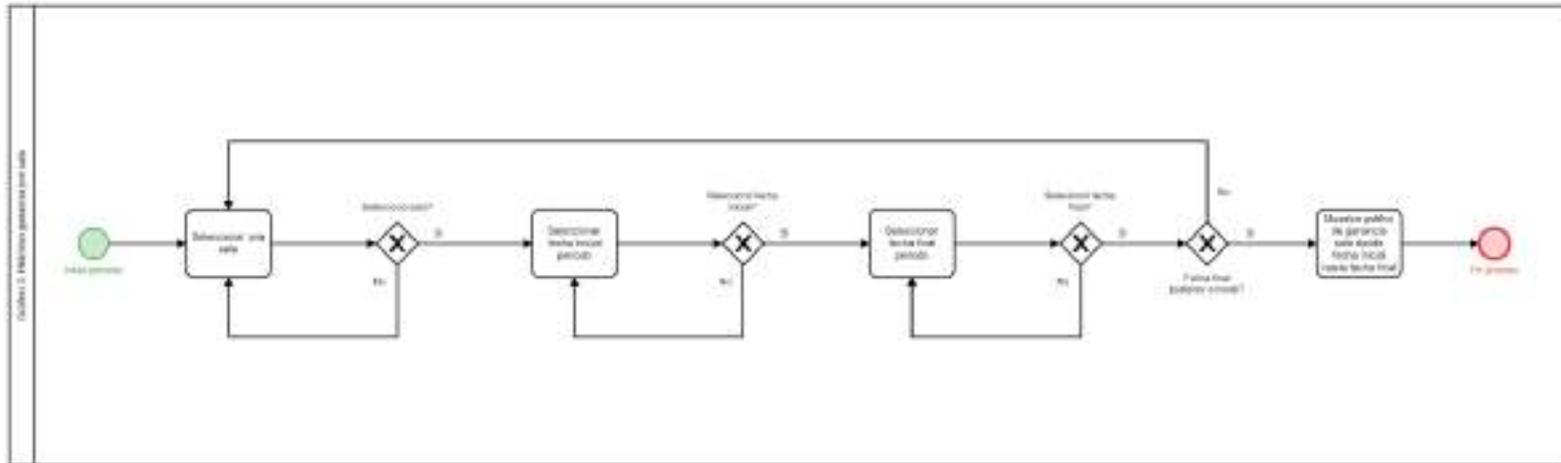
		seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	6	El sistema lanza un Alert de error indicando al usuario que las fechas ingresadas son erróneas.

Caso de Uso	#17. Visualizar Gráfico Estadístico 4: Ganancia Por Sala y Máquina en Fechas	
Actores	Gerente, Encargado, Empleado	
Propósito	Ver el gráfico estadístico 4 en la pestaña Datos	
Descripción	El usuario Gerente/Encargado/Empleado desea observar el gráfico 4 de la pestaña Datos, asociado a la ganancia por día de una máquina y sala en gráfico lineal en un rango de fechas especificadas.	
Precondiciones	Existen datos en la base de datos de movimientos de dinero en la sala y máquina.	
Curso Normal de los Eventos	Paso	Acción
	1	El usuario accede a la página principal del sistema.
	2	El usuario selecciona la pestaña Datos dentro del sistema.
	3	El sistema devuelve la página de Datos, donde el gráfico 3 se encuentra en la zona media de la página.
	4	El usuario completa Sala, Máquina, Fecha Inicial y Fecha Final.
	5	El usuario selecciona el botón Ver Gráfico.
	6	El sistema valida datos ingresados y muestra gráfico de ganancia por día desde Fecha Inicial hasta Fecha Final de la sala y máquina seleccionada.
Cursos Alternos	Paso	Evento
	2	El usuario no puede acceder a la página Mapa, ya que perdió su conexión a Internet al seleccionar dicha pestaña.
	5	El sistema lanza un Alert de error indicando al usuario que debe completar todos los campos requeridos.
	6	El sistema lanza un Alert de error indicando al

		usuario que las fechas ingresadas son erróneas.
--	--	---

Anexo III: Procesos operativos del sistema





Anexo IV: Manual de Usuario

Objetivos de la aplicación

Permitir el ingreso de usuarios existentes en el sistema ERP, crear y visualizar mapas de calor de un sector de una sala con la distribución de las máquinas tragamonedas en la misma en un rango de fechas especificadas, observar datos estadísticos en forma de gráfico, consultar las salas disponibles y usuarios del sistema.

Requisitos del sistema

Para poder utilizar el sistema son necesarios los siguientes requerimientos:

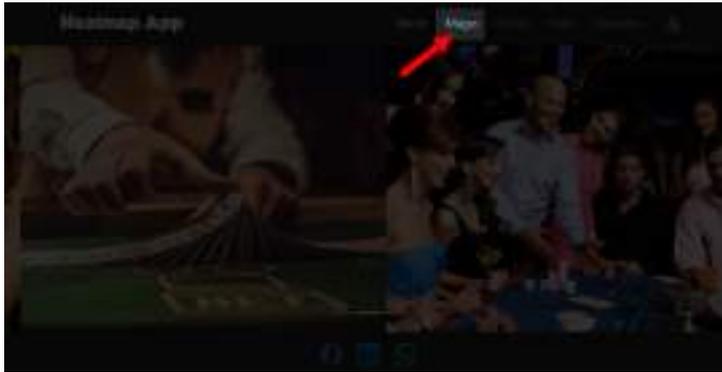
- Conexión a Internet con un proveedor.
- Disponer de un navegador (Chrome, Edge, Mozilla Firefox, Safari, etc.)

Ingreso

- 1) Abrir un navegador de preferencia.
- 2) Dirigirse a la URL: <https://www.heatmap10.herokuapp.com>
- 3) Identificarse con mismo usuario y contraseña usado en sistema ERP.
- 4) Si los datos son válidos se accede al sistema.

Crear mapa de calor

- 1) Seleccionar pestaña “Mapa” dentro de la aplicación



2) Clickear botón “Crear Mapa” (abre modal)



3) Seleccionar una sala ⁽¹⁾

4) Seleccionar un sector de la sala ⁽²⁾

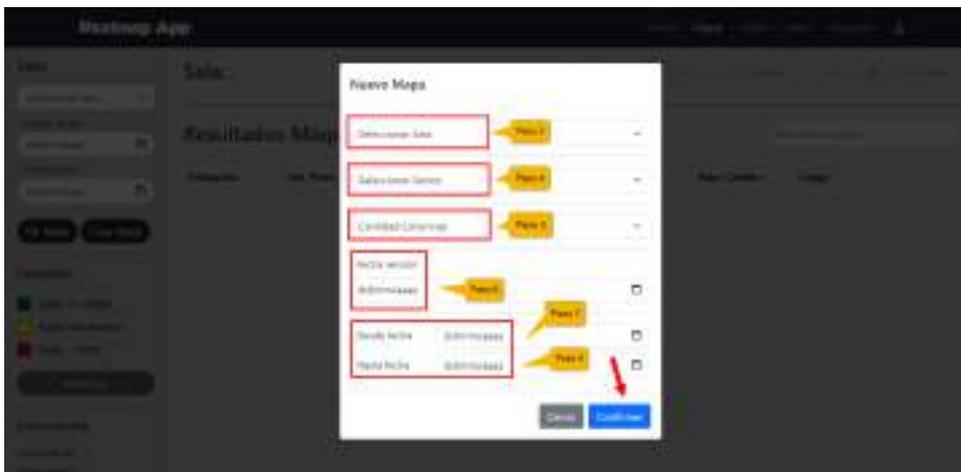
5) Seleccionar cantidad de columnas (opcional) ⁽³⁾

6) Seleccionar una fecha de versión ⁽⁴⁾

7) Seleccionar fecha inicial del período

8) Seleccionar fecha final del período ⁽⁵⁾

9) Clickear botón “Confirmar” (cierra modal) ⁽⁶⁾



10) Completar mapa con máquinas:

- Botón “ID Máquina”



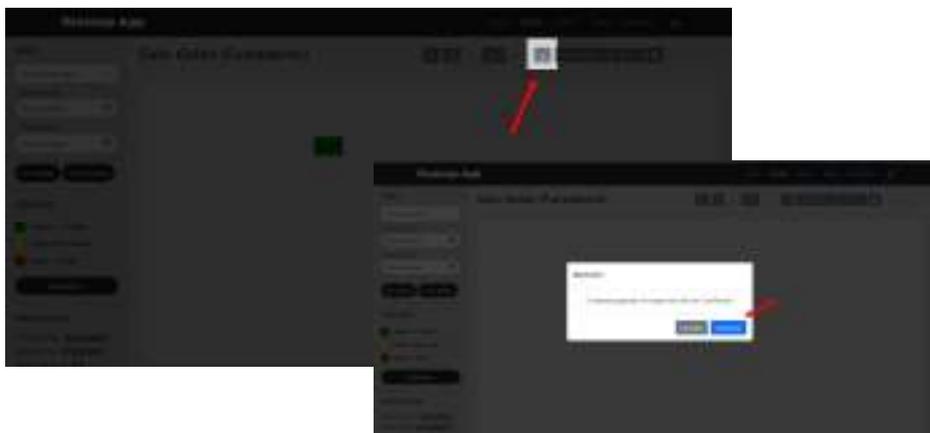
- Seleccionar máquina ⁽⁷⁾



- Posicionar máquina en lugar correspondiente usando botón "Libre"



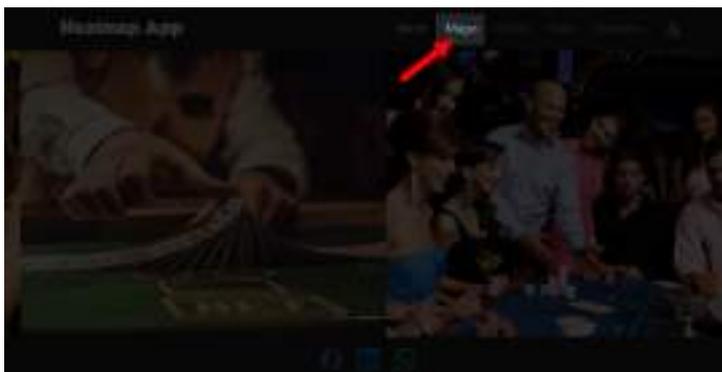
11) Guardar mapa con botón con símbolo de flecha



- ⁽¹⁾ Las salas propuestas son solo las que tienen resultados en la tabla resultados_slots de la base de datos MySQL.
- ⁽²⁾ Los sectores propuestos son los asociados a la salas según la tabla sala_sectores.
- ⁽³⁾ La cantidad de columnas indica la cantidad de máquinas que entran en el ancho del mapa. Luego la altura será definida en base al ancho para que puedan entrar todas las máquinas de la sala.
- ⁽⁴⁾ La fecha de versión será la fecha asociada a la creación de ese mapa.
- ⁽⁵⁾ La fecha final debe ser mayor a la fecha inicial. Caso contrario, lanza un error y limpia campos.
- ⁽⁶⁾ Si no existen resultados en las fechas asignadas, no se mostrará ninguna máquina para crear el mapa.
- ⁽⁷⁾ Las máquinas propuestas son solo las que tienen resultados en la sala indicada y en la tabla resultados_slots de la base de datos MySQL.

Ver mapa de calor

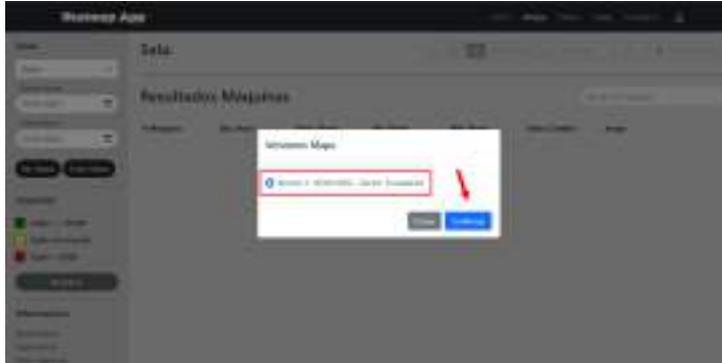
- 1) Seleccionar pestaña “Mapa” dentro de la aplicación



- 2) Seleccionar una sala ⁽¹⁾
- 3) Seleccionar fecha inicial del período
- 4) Seleccionar fecha final del período ⁽²⁾
- 5) Clickear botón “Ver Mapa” (abre modal) ⁽³⁾



- 6) Seleccionar versión
- 7) Clickear botón “Confirmar”



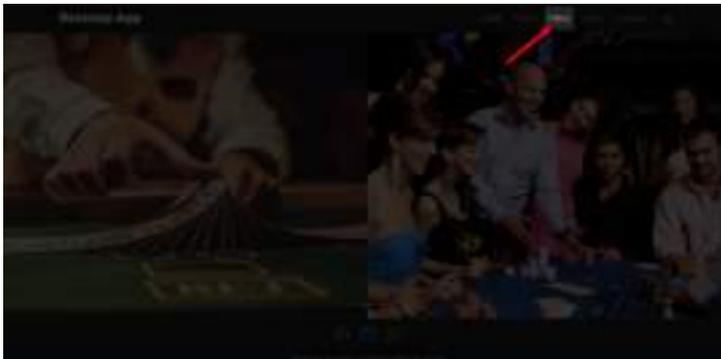
⁽¹⁾ Las salas propuestas son solo las que tienen resultados en la tabla resultados_slots de la base de datos MySQL.

⁽²⁾ La fecha final debe ser mayor a la fecha inicial. Caso contrario, lanza un error y limpia campos.

⁽³⁾ Si no existen resultados en las fechas asignadas, no se mostrará ninguna versión para visualizar.

Ver gráfico histórico ganancia por sala

- 1) Seleccionar pestaña “Datos” dentro de la aplicación



- 2) Seleccionar una sala ⁽¹⁾
- 3) Seleccionar una fecha inicial del período
- 4) Seleccionar una fecha final del período ⁽²⁾



5) Clickear botón “Ver Gráfico” ⁽³⁾

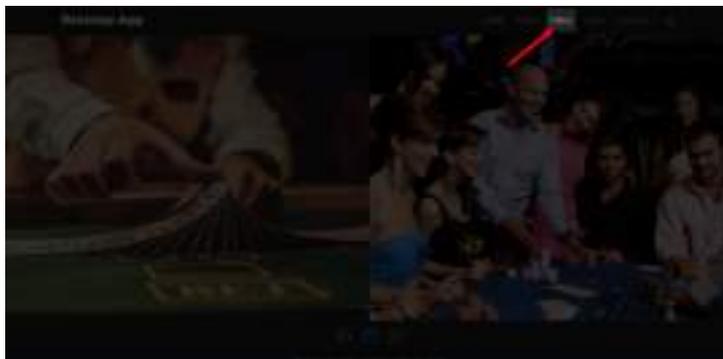
⁽¹⁾ Las salas propuestas son solo las que tienen resultados en la tabla resultados_slots de la base de datos MySQL.

⁽²⁾ La fecha final debe ser mayor a la fecha inicial. Caso contrario, lanza un error y limpia campos.

⁽³⁾ Si no existen resultados en las fechas asignadas, no se mostrará ningún gráfico asociado.

Ver gráfico histórico ganancia por máquina

1) Seleccionar pestaña “Datos” dentro de la aplicación



2) Seleccionar una sala ⁽¹⁾

3) Seleccionar una máquina de la sala ⁽²⁾

4) Seleccionar una fecha inicial del período

5) Seleccionar una fecha final del período ⁽³⁾

6) Clickear botón “Ver Gráfico” ⁽⁴⁾



- ⁽¹⁾ Las salas propuestas son solo las que tienen resultados en la tabla resultados_slots de la base de datos MySQL.
- ⁽²⁾ Las máquinas propuestas son solo las que tienen resultados en la sala indicada y en la tabla resultados_slots de la base de datos MySQL.
- ⁽³⁾ La fecha final debe ser mayor a la fecha inicial. Caso contrario, lanza un error y limpia campos.
- ⁽⁴⁾ Si no existen resultados en las fechas asignadas, no se mostrará ningún gráfico asociado.

Anexo V: Carta de conformidad del sistema

Por la presente nota se deja constancia que el producto desarrollado enmarcado en el Proyecto de Trabajo Final “Sistema de Mapas de Calor Aplicados a Movimientos Monetarios en Salas de Casino” llevado a cabo por el estudiante Lautaro Echeverría Luchini cumple las expectativas de la empresa Neogame S.A.

El Sr. Pablo Safronchik, referente funcional del Proyecto, deja por lo tanto su aval del Trabajo Final a entregar a la Cátedra en cuestión.

Mar del Plata, 23 de marzo del 2022

Pablo Safronchik

Firmado digitalmente