Marco Teórico y justificación.

Sistema de reservación de salones de eventos.

Se pretende desarrollar un software que puede ser aplicado como una herramienta útil para la reservación de salones arrendados para eventos. Este documento se estará enmarcando definir la metodología de desarrollo del sistema que permita a los desarrolladores o usuarios seguir obtener algunas especificaciones en cada una de las etapas de del desarrollo del sistema desde los requerimientos iniciales hasta las pruebas finales, que haga que el software sea coherente y además y le dé formalidad al mismo.

A continuación se abordaran los conceptos computacionales tomados en cuenta durante la elaboración del software de este proyecto nombrado (Resa).

Autor: Ezequiel Garcia Mancebo.

* Ingeniería del software

El establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales.

La Ingeniería del Software incluye la aplicación práctica del conocimiento científico en el diseño y construcción de los programas y la documentación requerida para su desarrollo, operación y mantenimiento.

* La ingeniería del Software abarca los siguientes puntos.
* Arquitectura del Software.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.

* Construcción de software.

El término Software Construction (Construcción del software) se refiere a la creación de software productivo y significativo a través de los procesos decodificación, verificación, pruebas unitarias, pruebas de integración y depuración de errores.

* Prueba de software.

Demostrar al desarrollador y al cliente que el software alcanza sus requisitos.

* Documentación.

La documentación de sistemas es el conjunto de información que nos dicen que hacen los sistemas como lo hacen y para quien lo hace.

La documentación adecuada y completa, de una aplicación que se desea implantar, mantener y actualizar en forma satisfactoria.

* Arquitectura de software.

Una Arquitectura de Software, también denominada *Arquitectura lógica*, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.

Tipos de arquitectura

**Descomposición Modular**. Donde el software se estructura en grupos funcionales muy acoplados.

**Cliente-servidor.** Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.

**Arquitectura de tres niveles**. Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente

* Programación por capas

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordiales la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

* Plataforma

En [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica), una plataforma es un sistema que sirve como base para hacer funcionar determinados módulos de [hardware](https://es.wikipedia.org/wiki/Hardware) o de [software](https://es.wikipedia.org/wiki/Software) con los que es [compatible](https://es.wikipedia.org/wiki/Compatibilidad_(inform%C3%A1tica)). Dicho sistema está definido por un estándar alrededor del cual se determina una [arquitectura de hardware](https://es.wikipedia.org/w/index.php?title=Arquitectura_de_hardware&action=edit&redlink=1) y una [plataforma de software](https://es.wikipedia.org/wiki/Plataforma_de_software)(incluyendo [entornos de aplicaciones](https://es.wikipedia.org/w/index.php?title=Entorno_de_aplicaciones&action=edit&redlink=1)). Al definir plataformas se establecen los tipos de [arquitectura](https://es.wikipedia.org/wiki/Arquitectura_(inform%C3%A1tica)), [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo), [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) o [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) compatibles.

* Ambiente \*-\*

Un entorno es un espacio o escenario informático en donde operan determinados comandos, funciones o características comunes.

* Infraestructura \*-\*

Es el conjunto de [hardware](http://es.wikipedia.org/wiki/Hardware) y [software](http://es.wikipedia.org/wiki/Software) sobre el que se asientan los diferentes sistemas.

* UI o (user interface)

La interfaz de usuario es el medio con que el usuario puede comunicarse o interactuar con una máquina, equipo, [computadora](https://es.wikipedia.org/wiki/Computadora) o dispositivo, y comprende todos los puntos de contacto entre el [usuario](https://es.wikipedia.org/wiki/Usuario) y el equipo.

* GUI user graphics interface

La interfaz gráfica de usuario, conocida también como GUI (del [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s) graphical user interface), es un [programa informático](https://es.wikipedia.org/wiki/Programa_inform%C3%A1tico) que actúa de [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario), utilizando un conjunto de imágenes y [objetos gráficos](https://es.wikipedia.org/wiki/Icono) para representar la información y acciones disponibles en la interfaz.

* GDI Graphics device interface

Graphics Device Interface (GDI) es uno de los tres componentes o subsistemas de la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) de [Microsoft Windows](https://es.wikipedia.org/wiki/Microsoft_Windows). Trabaja junto con el [núcleo](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)) y la [API de Windows](https://es.wikipedia.org/wiki/API_de_Windows).

Esta [interfaz de programación de aplicaciones](https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones) se encarga del control gráfico de los dispositivos de salida, como los [monitores](https://es.wikipedia.org/wiki/Monitor_de_computadora) o las [impresoras](https://es.wikipedia.org/wiki/Impresora).

* Temas o themes

Un tema (skin, piel en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s)), también llamado theme, tema o tapiz, es una serie de elementos gráficos que, al aplicarse sobre un determinado [software](https://es.wikipedia.org/wiki/Software), modifican su apariencia externa.

* UX (User xperience)

La experiencia de usuario es el conjunto de factores y elementos relativos a la interacción del [usuario](https://es.wikipedia.org/wiki/Usuario_(inform%C3%A1tica)), con un entorno o dispositivo concretos, cuyo resultado es la generación de una percepción positiva o negativa de dicho servicio, producto o dispositivo.

* Lenguajes de programación

Un lenguaje de programación no es más que un sistema estructurado y diseñado principalmente para que las máquinas y computadoras se entiendan entre sí y con nosotros, los humanos. Contiene un conjunto de acciones consecutivas que el ordenador debe ejecutar.

* Generaciones de los lenguajes de programación.

Los lenguajes de programación se dividen en 2 categorías fundamentales:

Bajo nivel: Son dependientes de la máquina, están diseñados para ejecutarse en una determinada computadora. A esta categoría pertenecen las 2 primeras generaciones.

Alto Nivel: Son independientes de la máquina y se pueden utilizar en una variedad de computadoras. Pertenecen a esta categoría la tercera y la cuarta generación. Los lenguajes de más alto nivel no ofrecen necesariamente mayores capacidades de programación, pero si ofrecen una *interacción programador/computadora más avanzada*. Cuanto más alto es el nivel del lenguaje, más sencillo es comprenderlo y utilizarlo.

Cada generación de lenguajes es más fácil de usar y más parecida a un lenguaje natural que su predecesores. Los lenguajes posteriores a la cuarta generación se conocen como *lenguajes de muy alto nivel*. Son lenguajes de muy alto nivel los generadores de aplicaciones y los naturales.

En cada nuevo nivel se requieren menos instrucciones para indicar a la computadora que efectúe una tarea particular. Pero los lenguajes de alto nivel son sólo una ayuda para el programador. Un mayor nivel significa que son necesarios menos comandos, debido a que cada comando o mandato de alto nivel reemplaza muchas instrucciones de nivel inferior.

1. Primera - Lenguaje de máquina: Empieza en los años 1940-1950. Consistía en sucesiones de dígitos binarios. Todas las instrucciones y mandatos se escribían valiéndose de cadenas de estos dígitos. Aún en la actualidad, es el único lenguaje interno que entiende la computadora; los programas se escriben en lenguajes de mayor nivel y se traducen a lenguaje de máquina.

2. Segunda - Lenguajes ensambladores: Fines de los ´50. Se diferencian de los lenguajes de máquina en que en lugar de usar códigos binarios, las instrucciones se representan con símbolos fáciles de reconocer, conocidos como *mnemotécnicos*,. Aún se utilizan estos lenguajes cuando interesa un nivel máximo de eficiencia en la ejecución o cuando se requieren manipulaciones intrincadas. Al igual que los lenguajes de la máquina, los lenguajes ensambladores son únicos para una computadora particular. Esta dependencia de la computadora los hace ser lenguajes de bajo nivel.

3. Tercera: años ´60. Los lenguajes de esta generación se dividen en tres categorías, según se orienten a:

Procedimientos: Requieren que la codificación de las instrucciones se haga en la secuencia en que se deben ejecutar para solucionar el problema. A su vez se clasifican en científicos (p.ej.: FORTRAN), empresariales (v.g.: COBOL), y de uso general o múltiple (p.ej.: BASIC). Todos estos lenguajes permiten señalar *cómo* se debe efectuar una tarea a un nivel mayor que en los lenguajes ensambladores. Hacen énfasis los procedimientos o las matemáticas implícitas, es decir en *lo* que se hace (la acción).

Problemas: Están diseñados para resolver un conjunto particular de problemas y no requieren el detalle de la programación que los lenguajes orientados a procedimientos. Hacen hincapié en la entrada y la salida deseadas.

4. Cuarta: su característica distintiva es el énfasis en especificar *qué* es lo que se debe hacer, en vez de como ejecutar una tarea. Las especificaciones de los programas se desarrollan a un más alto nivel que en los lenguajes de la generación anterior. La característica distintiva es ajena a los procedimientos, el programador no tiene que especificar cada pasó para terminar una tarea o procesamiento. Las características generales de los lenguajes de cuarta generación son:

* Threads o hilos.

Un lenguaje de programación no es más que un sistema estructurado y diseñado principalmente para que las máquinas y computadoras se entiendan entre sí y con nosotros, los humanos. Contiene un conjunto de acciones consecutivas que el ordenador debe ejecutar.

* Base de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

### Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

### Según la variabilidad de la base de datos

#### Bases de datos estáticas

### Son bases de datos únicamente de lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para [inteligencia empresarial](https://es.wikipedia.org/wiki/Inteligencia_empresarial).

#### Bases de datos dinámicas

### Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

### Según el contenido

#### Bases de datos bibliográficas

#### Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

#### Bases de datos o "bibliotecas" de información química o biológica

#### Son bases de datos que almacenan diferentes tipos de información proveniente de la [química](https://es.wikipedia.org/wiki/Qu%C3%ADmica), las [ciencias de la vida](https://es.wikipedia.org/wiki/Ciencias_de_la_vida) o [médicas](https://es.wikipedia.org/wiki/Medicina).

## Modelos de bases de datos

## Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores.

## Bases de datos jerárquica

En este modelo los datos se organizan en forma de árbol invertido (algunos dicen raíz), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

### Base de datos de red

Éste es un modelo ligeramente distinto del jerárquico, su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos, pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

### Bases de datos transaccionales

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

### Bases de datos relacionales

Éste es el modelo utilizado en la actualidad para representar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en [1970](https://es.wikipedia.org/wiki/1970) por [Edgar Frank Codd](https://es.wikipedia.org/wiki/Edgar_Frank_Codd), de los laboratorios [IBM](https://es.wikipedia.org/wiki/IBM) en [San José (California)](https://es.wikipedia.org/wiki/San_Jos%C3%A9_(California)), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "[tuplas](https://es.wikipedia.org/wiki/Tupla)". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por [registros](https://es.wikipedia.org/wiki/Registro_(base_de_datos)) (las filas de una tabla), que representarían las tuplas, y [campos](https://es.wikipedia.org/wiki/Campo_(base_de_datos)) (las columnas de una tabla).

### Bases de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de [Cubos OLAP](https://es.wikipedia.org/wiki/Cubo_OLAP). Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean aprender.

### Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los [modelos informáticos orientados a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos.

### Bases de datos documentales

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes, sirven para almacenar grandes volúmenes de información de antecedentes históricos. Tesaurus es un sistema de índices optimizado para este tipo de bases de datos.

### Bases de datos deductivas

Un sistema de base de datos deductiva, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática. Este tipo de base de datos surge debido a las limitaciones de la Base de Datos Relacional de responder a consultas recursivas y de deducir relaciones indirectas de los datos almacenados en la base de datos.

**SQL**

En la actualidad SQL es el estándar de facto de la inmensa mayoría de los SGBD comerciales. Y, aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

**Inyección SQL**

Inyección SQL es un método de infiltración de código intruso que se vale de una [vulnerabilidad informática](https://es.wikipedia.org/wiki/Error_de_software) presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una [base de datos](https://es.wikipedia.org/wiki/Base_de_datos).

El origen de la vulnerabilidad radica en el incorrecto chequeo o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código [SQL](https://es.wikipedia.org/wiki/SQL). Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) o [script](https://es.wikipedia.org/wiki/Lenguaje_interpretado) que esté embebido dentro de otro.

Justificación

En muchos centros públicos e instituciones privadas hoy en día tienen que cargar con el problema de no poder gestionar los salones de los cuales disponen de una manera efectiva y automatizada, con el fin de poder manipular los itinerarios e manejar los eventos programados en el mismo.

El desarrollo de este proyecto permitirá a esta comunidad comunidad empresarial:

* Automatización de las solicitudes a eventos.
* Control de eventos en los salones manipulación y candelarizacion.
* Control de reportes.
* En relación con los solicitantes tendrán la oportunidad de ser notificados en cuanto se apruebe su evento.

Con la implementación de este proyecto se busca beneficiar directamente a

la comunidad educativa la cual en muchos de los establecimientos no tienen un control de los salones, se espera que la puesta en práctica del mismo brinda una solución concreta del problema propuesto como objetivo de solución.