**Software de administración para la empresa**

**JVR Producciones**



Nu-CaBuPa. Software dedicado

Integrantes:

* Milagros Núñez
* Mateo Cabral
* Manuel Buslón
* Nahuel Pacheco

2020 Proyecto Integrador

Índice:

* Introducción……………………………………………………
* Fundamentación………………………………………………
* Nuestra empresa
* Contrato SRL
* Análisis F.O.D.A
* Primera reunión:
  + Entrevista.
  + Análisis.
* Análisis de requerimientos.
* Base de datos:
  + MER
  + Pasaje a tablas
  + Pasaje a sentencias de MySql
  + Creación de usuarios MySql
* Implementación del manual.
* Código actual.

Introducción:

Somos estudiantes de la "Escuela Superior Catalina Harriague de Castaños" procedentes del bachillerato E.M.T Informática (3er año) cuyo deber es crear una empresa dedicada al diseño y desarrollo de software con el propósito de cumplir con el proyecto de final de curso.

Dicho proyecto será desempeñado utilizando los conocimientos adquiridos en los años previos de estudio en el bachillerato de informática. También posee variedad de materias involucradas, las cuales son: Programación, Proyecto, Análisis y Desarrollo de Aplicaciones, Base de Datos, Formación Empresarial y Taller.

Nuestra empresa se definiría bajo el nombre “Nu-CaBuPa.SRL” y tendría su base en Uruguay, en el departamento de Salto. La misma se desarrollaría en el área tecnológica entorno a la producción de software dedicado desde fines administrativos hasta aplicaciones de entretenimiento.

El programa debe ser solo y únicamente para una empresa que necesite de este y pueda facilitarle tareas, las cuales serán indicadas por la empresa misma al igual que la organización del programa en caso de que el cliente desee especificarlo.

Nuestro software será diseñado para la empresa "JVR Producciones" y su propósito será enfocado en la administración y control de datos.

## Fundamentación:

A instancias de la necesidad por parte de las empresas de optimizar sus actividades han surgido proyectos de software de forma masiva. Cada uno adecuándose a las necesidades de un público específico o a un particular.

Este proyecto surge con los mismos fundamentos sobre los cuales emergen conjuntamente los demás, la posibilidad de volver más eficiente cualquier control e interacción entre los funcionarios y la gran cantidad de información a la que se enfrentan, en conjunto con las herramientas que estos deben utilizar continuamente.

Este software que será desarrollado, más que permitir una reducción en recursos y tiempo utilizado, lograrían tener una visión más amplia y objetiva de todas las actividades en las que se desarrolla la empresa. Se lo puede considerar directamente como una inversión realizada en el momento que se empieza el desarrollo, siguiendo esta visión, el software estará manteniendo una mejora continua en el tiempo sobre la utilización de recursos y gastos que gracias a este ya se volverían innecesarios.

En el sector de administración de recursos, lograría una automatización en la contabilidad de ganancias y un registro de las fuentes de ingreso en cada sector en la que se desarrolle. Permitiendo a dicho cliente un ahorro masivo en recursos tan valiosos como puede serlo el tiempo. Un ejemplo de este punto puede ser el acceso a variada cantidad de información almacenada ahora en servidores, pudiendo realizar consultas en un tiempo exponencialmente menor que el utilizado anteriormente. Una consecuencia directa de este ejemplo es la visibilidad de dicha información para el personal autorizado a acceder a ella, sin trabas externas a la empresa debido a las herramientas y la posibilidad de disminuir casi totalmente la perdida de información o destrucción de la misma por errores debido al método de almacenamiento empleado.

En conjunto al avance del tiempo y las tecnologías más actuales que se empleará a futuro el software contaría con la posibilidad de evolucionar y adaptarse al contexto, también contar con requisitos futuros en los que se deberá desempeñar. Dicha información almacenada y estructura en la codificación está regida por los paradigmas de la programación empleados en el mercado actual. Trayendo esto consigo una gran posibilidad de migración a diferentes software, remodelado de apartados en el programa y exportación de datos.

Como filosofía de este proyecto tenemos la utilización de una amplia gama de conocimientos, patrones de diseño y algoritmos estudiados y probados en entornos para el perfeccionamiento del programa en gran cantidad de casos que podrían ocurrir. Sumado a la correcta realización de pruebas y aplicación de diferentes herramientas en el desarrollo el software, contaría con el mejor desempeño y optimización adecuado a las necesidades del cliente. Todo esto permitiendo cambios futuros del mismo para obtener un mejor desempeño en las áreas que sean requeridas.

# Nuestra empresa:

Misión:

La empresa de software denominada Nu-CaBuPa.SRL está comprometida con el cumplir de las necesidades y deseos de los clientes, tanto como el análisis de procedimientos aplicados en búsqueda de optimizaciones e implementaciones de las mismas. Todo a manera de seguir brindando productos de calidad, cumpliendo con las expectativas de los clientes y el nivel exigido por el mercado. Brindando nuevos métodos y tecnologías, y sucesivamente, competitividad.

Visión:

Como empresa deseamos alcanzar un nivel de referencia internacional en cuanto a tecnología se refiere. Solucionar los problemas que el mundo tecnológico presente y crear nuevos estándares. Mantenernos actualizados a los últimos procedimientos informáticos e indirectamente iniciando nuevos avances debido a la necesidad del cambio. Siendo reconocida no por la velocidad, ni cantidad, si no calidad y el impacto.

Objetivos:

-Hacer reconocidos nuestros esfuerzos y logros.

-Optimizar el tiempo de producción en un 24%.

-Mejorar la eficiencia de los programas y tecnologías utilizadas con cada nuevo proyecto.

-Perfeccionar la comunicación entre los integrantes.

-Fomentar un área de trabajo saludable y participativa entre todo el personal.

-Expandirse a otros mercados en el sector de software.

-Diseñar un modelo empresarial rentable para el futuro.

-Tener una estrategia de marketing en caso de ser necesario.

-Atraer más socios capacitados que ayuden a la empresa y traigan su propia visión con ellos, a futuro.

S.R.L.

**SOCIEDADES DE RESPONSABILIDAD LIMITADA**

*Son sociedades que solamente pueden hacerse a medida y por encargo de cada cliente, con por lo menos dos socios.*

* Los socios no deberían tener embargos inscriptos.
* La Administración estará a cargo de los socios en la forma que ellos mismos acuerden, por ejemplo en forma conjunta o indistinta, etc.
* El giro puede ser el estándar y particularmente se incluye como principal, la actividad más importante que se va a realizar.
* Las Sociedades se entregan inscriptas en DGI y BPS, debiéndose establecer para BPS, cual es el socio que aporta y que va a tener actividad.
* Los aportes a BPS se pueden realizar por uno solo de los socios, que deberá ser el socio que declare actividad.
* Los socios serán responsables de las deudas sociales, hasta el máximo del capital social, respondiendo con sus propios bienes por las obligaciones de la sociedad.
* Los socios permanecerán en la sociedad y solo por cesión de sus cuotas sociales podrán dejar de participar en la misma. Las cesiones de cutas también se inscriben en el registro nacional de comercio y se publican en Diario Oficial y particular.

**EJEMPLO DE CONTRATO DE SOCIEDAD DE RESPONSABILIDAD LIMITADA.**

PRIMERO. (Denominación) La naturaleza de la sociedad será de responsabilidad limitada, para la que adoptan la denominación de Nu-CaBuPa.SRL. Se regirá por la Ley 16.060, el decreto 155/010 de 24 de mayo de 2010, y demás normas acordadas

SEGUNDO. (Domicilio) La sociedad tendrá su domicilio en el departamento de Salto, pudiendo establecer sucursales, filiales, agencias o representaciones en todo el país o en el extranjero.

TERCERO. (Plazo) El plazo de duración de la sociedad será de 8 años a contar de hoy, no obstante, cada año, a contar desde hoy, cualesquiera de los socios podrá hacer uso del derecho de renuncia, debiendo en tal caso hacer saber su decisión con una anticipación no menor de tres meses al vencimiento del período respectivo, a los demás socios por telegrama colacionado.

CUARTO. (Objeto) La sociedad tendrá por objeto la realización de la siguiente actividad: A) Desarrollo de Software. B) Gestión, administración, soporte de los productos desarrollados por nuestra empresa; C) Ayuda técnica con el uso del sistema informático D) realización de toda clase de actos y/u operaciones civiles, comerciales, industriales, ya sea por cuenta propia, de terceros o de ambos a la vez, pudiendo a tales efectos comprar, vender, hipotecar, caucionar, administrar, explotar, arrendar y dar en arrendamiento, cualquier clase de bienes y derechos, sean muebles, inmuebles, corporales o incorporales, pudiendo a tal efecto participar en toda clase de sociedades y empresas existentes o que se constituyan en el futuro. La sociedad podrá realizar si lo estima conveniente para el mejor desarrollo de la empresa o para un integral aprovechamiento de su capital, cualquier acto, negocio, o contrato de cualquier naturaleza y de cualquier ramo de actividad comercial. Como medios adecuados para el cumplimiento de sus objetivos, la sociedad podrá realizar todos los actos jurídicos referidos para el desarrollo de sus negocios, tanto sean de disposición, afectación y administración, otorgando y efectuando cualquier clase de operaciones ya fueran de enajenación, adquisición, afectación y gravámenes sobre toda clase de bienes y derechos, sin limitación alguna, pudiendo al efecto participar en toda clase de sociedades y empresas existentes o que se constituyan en el futuro; E) Implementación, diseño, compra, venta de equipos informáticos y sistemas de redes de ordenadores; F) Desarrollo de contenido audio visual con fines de entrenamiento; G) Traducción y exposición de cursos y conferencias en relación al ámbito tecnológico-psicológico-filosófico-matemático; H) Instalación y obtención de software relacionado a herramientas de edición de texto.

QUINTO. (Capital) El capital de la sociedad se fija en la suma de $288.640 (pesos uruguayos que quedan divididos en 40 cuotas)

Cuotas de $7.216 cada una, correspondiéndole a los socios Sres.

La sociedad estará conformada por: Manuel Buslón, Mateo Cabral, Milagros Nuñez, Nahuel Pacheco. Cada uno aportando 10 cuotas, equivalentes a $72.160, lo cual ya fue pagado.

SEXTO. (Aportes) Por su obligación de aportar, cada uno de los socios manifiesta que aportó a la sociedad antes de este acto y en efectivo, la suma correspondiente a sus cuotas, con lo que queda integrado en su totalidad el capital societario previsto, razón por la cual se otorgan recíprocas cartas de pago por sí y en representación de la sociedad.

SEPTIMO. (Calidad de los socios) La responsabilidad de los socios quedará limitada a la cantidad o cantidades estipuladas como aporte de capital en el contrato social. ————————————–

OCTAVO. (Administración) La administración de la sociedad y el uso de la firma social, con las más amplias facultades y con el ejercicio de la representación de la empresa estarán a cargo de cualquiera de las socias, actuando indistintamente, sin perjuicio de la facultad de otorgar mandatos a terceros. A título enunciativo se establece que el administrador en nombre y representación de la sociedad podrá: a) realizar toda clase de actos de disposición, administración y afectación quedando facultado para enajenar y/o gravar toda clase de bienes; b) celebrar toda clase de contratos; c) ejecutar todo género de actos de administración; d) otorgar todo tipo de mandatos; e) representar a la sociedad ante cualquier autoridad judicial o administrativa; f)suscribir todo tipo de documentos civiles y/o comerciales; g) registrar todo tipo de Marcas y Patentes. El administrador tendrá todas las facultades necesarias para el gobierno, administración y disposición de los bienes de la sociedad, así como también la representación judicial, extrajudicial y administrativa de la misma, y podrá actuar por sí o hacerse representar por apoderado con facultades suficientes. El domicilio del administrador será el indicado como suyo en la comparecencia de este contrato. El administrador deberá suscribir la documentación de la siguiente forma: por S.R.L. seguido de su firma habitual. ——————————————–

NOVENO. (Inventario-balance) Anualmente se efectuará un inventario-balance al cierre del ejercicio económico el que deberá estar concluido dentro de los ciento veinte (120) días siguientes a la finalización del ejercicio. Los socios tendrán las más amplias facultades para controlar la confección de los balances y formular por escrito las observaciones que les merezcan. Los socios acuerdan que el ejercicio económico finalice cuando lo establezca la asamblea de socios. ————

DECIMO. (Fondo de reserva.) De las ganancias líquidas de cada ejercicio se deducirá un diez por ciento (10%) para formar un fondo de reserva, hasta que éste alcance un cien por ciento (100%) del capital social. ———————————————————————————–

DECIMO PRIMERO. (Distribución de ganancias y pérdidas) El resto de las ganancias serán repartidas y las pérdidas soportadas, entre los socios en proporción de sus aportes. Las pérdidas de cada ejercicio serán compensadas con las utilidades del o de los ejercicios subsiguientes.—————————————————————

DECIMO SEGUNDO. En caso de ausencia, fallecimiento o incapacidad de cualquiera de los socios, la sociedad continuará entre los demás socios.——– ——————————————————————

DECIMO TERCERO. La sociedad se disolverá cuando así lo decida la Asamblea Extraordinaria convocada a tales efectos y será ésta quien decidirá la o las personas que realizarán la liquidación de los bienes sociales fijando sus atribuciones y remuneraciones. Una vez liquidados los bienes sociales del activo y abonadas todas las obligaciones de la sociedad, el remanente será adjudicado a los socios en proporción a sus respectivos capitales. La sociedad puede disolverse por las siguientes causales: A); Por decisión unánime de los socios; B) Por expiración del plazo; C) Por pérdidas que reduzcan el patrimonio social a una cifra inferior a la cuarta parte del capital social integrado; D) Por reducción a uno del número de socios según lo dispuesto en el art. 156 de la ley 16.060. ————————————————————

DECIMO CUARTO. (Cesión de cuotas sociales). Las cuotas sociales podrán ser cedidas por cualquiera de los socios conforme a las disposiciones legales. Cuando un socio desee ceder su cuota a un tercero ajeno a la sociedad, se convocará a una Asamblea Extraordinaria, diez (10) días después de que el socio interesado comunique por telegrama colacionado la intención de ceder su cuota. En tal caso la cesión deberá ser aceptada por unanimidad. Si no se notificara la oposición se presumirá el consentimiento. Si se formulara alguna oposición se seguirá el procedimiento previsto por el art. 232 de la ley 16.060. La sociedad tendrá prioridad frente a los socios para la adquisición de las cuotas. Si hubiera varios socios interesados en la adquisición, las cuotas se distribuirán a prorrata y si no fuera posible, se atribuirán por sorteo.———————————————————————————–

DECIMO QUINTO. Los socios podrán por unanimidad, fijarse remuneraciones en calidad de sueldo y otros conceptos, así como autorizar retiros a cuenta de utilidades con débito a sus respectivas cuentas. —————————————————————-

DECIMO SEXTO. Los socios se obligan muy especialmente a aportar su trabajo a los negocios de la sociedad, pudiendo no obstante realizar operaciones mercantiles por su cuenta o como integrante de otras empresas, siempre que su realización no perjudique a la sociedad, ni menoscabe la actividad que le corresponda desarrollar en la misma. —-

DECIMO SEPTIMO. Anualmente dentro de los ciento veinte (120) días de cerrado el ejercicio económico se celebrará una Asamblea Ordinaria que tendrá por objeto aprobar el balance, examinar y juzgar la gestión de la administración, así como la cuenta de ganancias y pérdidas, establecer el porcentaje destinado al capital de reserva cuando éste fuera mayor al fijado, y resolver cualquier otro punto que se encuentre en el orden del día. Los miembros de la Asamblea serán citados por telegrama colacionado, con una anticipación de por lo menos diez (10) días a la fecha de realización de la misma. La Asamblea Extraordinaria podrá ser convocada en cualquier momento, por mayoría de socios, debiendo comunicar por escrito a los restantes la realización de la misma con una antelación no menor a diez (10) días. El quórum necesario para sesionar y tomar resoluciones en Asambleas Ordinarias o Extraordinarias, salvo disposición en contrario, estará constituido por la mayoría de los socios que representen la mayoría del capital social, correspondiéndoles a tales efectos un voto por cuota social. ——————————————

DECIMO OCTAVO. Queda prohibido a la sociedad constituirse fiadora de los socios y de terceras personas. ———————————–

DECIMO NOVENO. La asamblea social, por decisión de la mayoría de socios que represente las tres cuartas partes del capital social podrá tomar las siguientes decisiones: a) transformar a la SRL en sociedad anónima; b) fusionarla con otra; c) modificar el objeto social; d) revocar y nombrar administradores; e) prorrogar el plazo de la sociedad. Los socios disidentes o ausentes tendrán derecho a receso. ———————————————————————————-

VIGESIMO. Los socios podrán renunciar a la sociedad en los siguientes casos: a) en caso establecido en la cláusula tercera de este contrato; b) si el ejercicio social diere una pérdida que excediere el 30% del capital social; c) si no se conformaren con la resolución que tomen los restantes socios respecto a lo previsto en la cláusula décimo novena de la presente. En tales casos la participación del socio saliente se determinará conforme al balance practicado del día de la manifestación de voluntad, y la cuota que correspondiere le será abonada por la sociedad hasta en cuatro mensualidades consecutivas, iguales a contar de la fecha del balance, las que devengarán el máximo interés legal, calculado sobre los saldos deudores pagadero conjuntamente con el capital. ———————————————————————————

VIGESIMO PRIMERO. Cualquier diferencia o controversia entre los socios, sea por aplicación o interpretación de este contrato, será resuelto inapelablemente por árbitros, designados uno por cada una de las partes discrepantes y un tercero designado por los árbitros ya designados. La designación se hará dentro del plazo de 30 días a contar del telegrama colacionado que el discrepante deberá enviar a la administración de la sociedad, manifestando su disconformidad. —–

VIGESIMO SEGUNDO. En todo lo que no estuviere previsto en este contrato se aplicarán las disposiciones de la ley 16.060 sección IV, “De las sociedades de responsabilidad limitadas” y sus concordantes.———

|  |  |  |
| --- | --- | --- |
|  | Interno | Externo |
| Ventaja: | Fortalezas:   * Conocimientos sobre desarrollo de aplicaciones y experiencia en el mismo. * Buena comunicación y organización interna. * Metas y objetivos claros. * Precios ajustados al producto y al trabajo requerido. * Clase y singularidad en los trabajos realizados. | Oportunidad:   * Facilidad de crecimiento como empresa debido al rubro. * Necesidad por parte de las empresas de aplicaciones y servicios. * Mercado objetivo claro. * Facilidad para distribución del producto. * Coste casi inexistente de producción de una vez finalizado el producto. |
| Desventajas: | Debilidades:   * No tener una fuerte relación previa con nuestros clientes. * Poca experiencia en el mercado. * Capital inicial inexistente. * Escasez de personal capacitado. * Incapacidad de aceptar muchos trabajos. | Amenazas:   * Alojamiento en un país tercermundista. * Gran competencia laboral. * Crisis mundial. * Tendencia a un ahorro de recursos y cuidados en las inversiones. * Trabajo y dependencia de terceros por el uso de sus productos. * Poco conocimiento del entorno sobre la factibilidad y posibilidad del software. |

# Entrevista:

Luego de la presentación de nuestro grupo, contando quiénes somos y lo que podemos ofrecer al cliente se dio la entrevista de requerimientos.

Grupo: ¿Puede contarnos qué tipo de actividades realiza su empresa?

Cliente: Principalmente la gestión y el control de la radio Bemba

Grupo: ¿Y cómo se estuvo administrando los datos de la radio hasta ahora?

Cliente: Tengo una libreta en el que anoto los pagos de cada programa y más o menos los horarios son sobre lo que se recuerda y lo que queda hablado con el dueño de cada programa. Las publicidades dependen de la tanda en que se contrataron y queda guardado lo que se debe decir o mostrar en la computadora

Grupo: ¿Y de esos programas le interesaría almacenar alguna información?

Cliente: Solo más o menos saber que conductores hay y los pagos de cada mes por parte del dueño del programa. También estaría el operador pero es casi siempre el mismo conductor el que hace esa función y actúa como el dueño.

Grupo: ¿Los precios para los programas están definidos antes o se acuerdan con el que alquilaría el espacio?

Cliente: Se fijan en el momento teniendo en cuenta lo que duraría, repeticiones por semana y de que trata

Grupo: Entonces ¿un programa puede tener varios días distintos cada semana?

Cliente: Sí, y eso puede cambiar

Grupo: ¿Y el tema de publicidades cómo es?

Cliente: En el momento de la tanda entre cada programa se comentan o muestran las publicidades que contratan las empresas, cada una con las repeticiones que pagó. De las publicidades que aparecen en cada programa no interesaría, ya que el dueño del programa se encarga de eso.

Grupo: De eso ¿solo le interesaría saber lo de los pagos, la tanda en que aparecen y las repeticiones entonces?

Cliente: Sí, me gustaría que quede guardado algo sobre publicidades que han sido contratadas y los pagos para luego calcular. ¿Se podrá?

Grupo: Sí, se podría a base de la información del pago de los programas y las publicidades hacer que el software calcule las ganancias de los meses o días que se quiera saber. ¿En la empresa quienes usarían el software?

Cliente: Principalmente yo y otro que esté ahí ayudando a registrar las cosas, el operador de la radio puede ser. También quisiera poder usarlo desde casa u otro lugar.

Grupo: Bueno, podemos hacer que se administre de forma online. ¿Tiene un servidor propio?

Cliente: Sí, tengo uno en el que está la página web de la radio y en el que se retransmite la misma

Grupo: ¿Qué equipos tiene? ¿Son equipos de uso propio o exclusivos para la administración de la radio?

Cliente: Mis equipos de administración son de alta eficiencia, en la radio las computadoras son exclusivos para los programas, mientras que la computadora que tengo en mi casa está más para el trabajo.

Grupo: Está bien, en una futura entrevista acordamos como implementamos eso y qué características tiene el servidor si se puede usar ese. ¿Hay algo más que quisiera mantener en los registros?

Cliente: Cada tanto hay eventos que se realizan con algún tema por la empresa, lo importante sería saber la fecha y de que trata más o menos. Después está lo de Mil voces y otras series que produciríamos, de eso es solo tener que video de la serie se va a emitir y de que trata.

Grupo: ¿Esos eventos tienen relación con algo de la radio o las series?

Cliente: Puede tratar de eso sí, estar en conjunto con algún programa sobre el que se realice o por algún video o tema de la serie para promocionarlo. En el evento se mostraría publicidades también.

Grupo: ¿Algo más que quiera poder administrar?

Cliente: Por ahora serían solo esas cosas, si pienso algo más les envío un mensaje.

Grupo: ¿Le gustaría mantener control de las publicidades que aparecerían en algún programa de la radio?

Eso dentro del software

Cliente: Sí, me gustaría, claro si es posible.

# Análisis de la entrevista:

La entrevista a nuestro cliente JVR Producciones fue realizada por medio de la plataforma zoom el día 06 del 04 por motivos de una emergencia sanitaria. Esta fue planteada con anticipación, planeando con el equipo nuestra presentación y una metodología, la cual nos serviría para la obtención de datos y que a raíz de eso generaríamos más preguntas para la recolección de todos los requerimientos deseados por el cliente para poder realizar el software a medida.

Las primeras preguntas que fueron realizadas dieron un enfoque a la obtención de información sobre la empresa, para la realización del programa nuestro equipo tiene que estar en conocimientos sobre las actividades que realiza nuestro cliente y como estuvo administrando los datos hasta el momento, para poder identificar los métodos que este tenía. Por medio de la respuesta del cliente pudimos establecer que sus métodos de administración son bastante ineficientes, ya que estos demandan una buena organización por parte del mismo, además de que este método utilizado puede generar retrasos, errores y posibles inconvenientes.

Al momento de la obtención de la información el equipo comenzó a realizar preguntas relacionadas con el interés del cliente que se basaban en la gestión del programa de tv y una administración de control con relación a la radio, para poder comenzar a sugerir ciertas propuestas para la realización del programa dando sugerencias para su creación final.

A medida que se iba obteniendo información, se comenzó a consultar sobre los requisitos del programa, realizando preguntas especificas que estuvieran conectadas con la información que el cliente nos brindaba. Preguntando cómo es que se hacían ciertas actividades al momento de la administración, como es que se realizaban esas actividades y haciendo un énfasis en el interés del cliente, dando sugerencias y algunas explicaciones al momento que este pedía algo en específico.

Para concluir con la recolección de información necesaria, se comenzó a preguntar sobre el entendimiento que el cliente tenía sobre el área, además de preguntar sobre quienes estarían utilizando el programa cuando esté finalizado, para poder saber cómo crear el manual de usuario, además de conocer cuáles eran los equipos que este manejaba, ya que el programa será utilizado tanto en la radio como en la casa del cliente.

Al finalizar se concordaron ciertas especificaciones para el diseño que el cliente optó por propias.

Análisis de los requerimientos:

|  |  |
| --- | --- |
| Definición de requisitos de usuario: | Especificación de requisitos del sistema: |
| El consumidor debe poder tener un control sobre los distintos usuarios y sus permisos. | El usuario poseería campos para seleccionar los distintos permisos que se tengan además de poder controlar los nombres y contraseñas (si posee los permisos necesarios). |

Usuarios:

Programa:

|  |  |
| --- | --- |
| El usuario podría administrar el precio de la cuota de mensual de cada programa. | Se podría elegir entra varias opciones de precios ya establecidos o agregar nuevos precios para otras categorías/programa. |
| El programa poseería fechas en las que se emiten y estas podrían ser controladas. | Se mostrarían en una lista las fechas ya establecidas y por medio de un botón se podría seleccionar en un calendario para agregar otra fecha.  Seleccionando una fecha de la lista se podría eliminar o modificar. |
| En el programa es posible que se muestren publicidades y se permitiría mantener el control sobre cuáles y hasta que fecha serían emitidas. | En una interfaz se podría seleccionar al programa que se desearía mostrar la publicidad y cuál de estas sería. |
| Se podría definir que un evento está relacionado con un programa. | Sería posible enlazarlo por medio de seleccionarlos de dos listas. |

Funcionarios:

|  |  |
| --- | --- |
| Se podría definir las funciones de cada funcionario. | A partir de seleccionar de un perfil del mismo dichas funciones (O agregarla si no se han creado con anterioridad). |
| El usuario puede establecer que funcionario y cumpliendo qué función tendría cada programa. | A base del funcionario y de la función se marca en que programa la ejercería. |

Publicidad:

|  |  |
| --- | --- |
| La publicidad tiene cuotas que se deben pagar y se registrarían las fechas. | Ingresando en el administrador de publicidades se pueden verificar las distintas cuotas a partir de una tabla, en la que se mostrar |
| Todas las publicidades pertenecen a alguna empresa y se debe poder tener registro de a cual. | El administrador de empresa se mostraría las diferentes publicidades pertenecientes de dicha empresa en el orden en el cual se agregaron.  Se permitiría realizar una búsqueda por tema. |
| En la radio hay distintas tandas en las que pueden aparecer publicidades. | Permitiría administrar las tandas junto a las publicidades que aparecerían en la misma. Junto con un control de la fecha que empezarían a emitirse, junto a la fecha de finalización. |
| En los eventos se muestran publicidades y sería conveniente tener un registro de las mismas. | Mediante la unión a bese de la selección en listas (o buscando por temas) se relacionarían la publicidad con el evento. |

Videos:

|  |  |
| --- | --- |
| El usuario tendría el control de los videos y de las series, junto con la pertenencia de los mismos a las series. | Una vez agregado el video (el nombre y el tema) se podría seleccionar una serie existente a la cual pertenecería el mismo. |
| Sería posible que un evento tratara de un video de una serie y sería conveniente conocer los datos de ambos. | En una interfaz del control del evento habría una tabla con los videos relacionados permitiendo añadir, modificar y eliminar los mismos. |

Base de datos

MER:

# Proyecto.png

|  |
| --- |
| Pasaje a tablas |
| Programa: { ID\_Programa , Nombre\_Programa, Fecha\_Finalizacion}    FechaPrograma: { Fecha , Hora\_Inicio , Hora\_Fin, ID\_Programa}  PrecioEspacio: { ID\_Precio\_Espacio , Descripcion, Precio}  ProgramaCuota: { ID\_Precio\_Espacio , ID\_Programa ,Fecha\_Pago,Nro\_Recibo}  Publicidad: { ID\_Publicidad , Tema, ID\_Empresa}    usuarios: { id\_usuario , nombre, contrasena}  acceso: { id\_acceso , seccion}  UsTieneAcceso: { id\_usuario, id\_acceso }    PMuestraPubli: { ID\_Publicidad , ID\_Programa , Fecha\_Finalizacion, Fecha\_Inicio}  PublicidadCuota: { ID\_Cuota , Fecha\_Pago, Precio, ID\_Publicidad}    Empresa: { ID\_Empresa , Nombre, Telefono}    Tanda: { Hora\_Inicio , Hora\_Fin}  AparecePubli: { ID\_Publicidad , Hora\_Inicio , Fecha\_Inicio, Fecha\_Finalizacion}    Evento: { ID\_Evento , Nombre, Descripcion}    VEsDeEvento: { ID\_Evento, ID\_Video}  FechaEvento: { ID\_Evento, Fecha }  EventoMuestraPubli: { ID\_Evento , ID\_Publicidad }  EventoEsDeP: { ID\_Programa , ID\_Evento }    Funcionario: { ID\_Funcionario , Telefono, Nombre}    Funcion: { ID\_Funcion , Descripcion}  TrabajaComo: { ID\_Funcionario, ID\_Funcion }  FunTrabaja: { ID\_Funcionario, ID\_Funcion, ID\_Programa , Fecha\_Inicio, Fecha\_Finalizacion}  Video: { ID\_Video , Contenido, Nombre, ID\_Serie, Fecha}  Serie: { ID\_Serie , Fecha\_Finalizacion, Nombre} |

|  |
| --- |
| Consultas de creación MySql |
| CREATE TABLE `usuarios` (  `id\_usuario` int(6) unsigned auto\_increment not null primary key,  `nombre` varchar(64) NOT NULL,  `contrasena` varbinary(256) NOT NULL  );  CREATE TABLE `acceso` (  `id\_acceso` int(6) unsigned auto\_increment not null primary key,  `seccion` varchar(64) NOT NULL  );  CREATE TABLE `UsTieneAcceso` (  `id\_usuario` int(6) unsigned not null,  `id\_acceso` int(6) unsigned not null,  PRIMARY KEY (`id\_usuario`,`id\_acceso`),  KEY `id\_acceso` (`id\_acceso`),  CONSTRAINT `UsTieneAcceso\_ibfk\_1` FOREIGN KEY (`id\_usuario`) REFERENCES `usuarios` (`id\_usuario`),  CONSTRAINT `UsTieneAcceso\_ibfk\_2` FOREIGN KEY (`id\_acceso`) REFERENCES `acceso` (`id\_acceso`)  );  create table Programa  (  ID\_Programa int(6) unsigned auto\_increment not null primary key primary key,  Nombre\_Programa varchar(64) not null,  Fecha\_Finalizacion datetime  );  ---  create table FechaPrograma  (  Fecha date,  Hora\_Inicio time not null,  Hora\_Fin time not null,  ID\_Programa int(6) unsigned,  foreign key ID\_Programa references Programa(ID\_Programa),  primary key(Fecha, Hora\_Inicio));  create table PrecioEspacio  (  ID\_Precio\_Espacio int(6) unsigned auto\_increment not null primary key,  Descripcion varchar(256) not null,  Precio int(5) unsigned  );  create table ProgramaCuota  (  ID\_Precio\_Espacio int(6) unsigned not null,  ID\_Programa int(6) unsigned not null,  Fecha\_Pago date,  Nro\_Recibo int(7) unsigned,  primary key(ID\_Programa,ID\_Precio\_Espacio),  foreign key ID\_Programa references Programa(ID\_Programa),  foreign key ID\_Precio\_Espacio references PrecioEspacio(ID\_Precio\_Espacio)  );  create table Empresa  (  ID\_Empresa int(6) unsigned auto\_increment not null primary key,  Nombre varchar(64) not null,  Telefono int(9) unsigned  );  ---  create table Publicidad  (  ID\_Publicidad int(6) unsigned auto\_increment not null primary key,  Tema varchar(256) not null,  ID\_Empresa int(6) unsigned not null,  foreign key ID\_Empresa references Empresa(ID\_Empresa)  );  ---  create table PMuestraPubli  (ID\_Publicidad int(6) unsigned not null,  ID\_Programa int(6) unsigned not null,  Fecha\_Finalizacion date,  Fecha\_Inicio date,  foreign key ID\_Programa references Programa(ID\_Programa),  foreign key ID\_Publicidad references Publicidad(ID\_Publicidad)  );  create table PublicidadCuota  (  ID\_Cuota int(6) unsigned auto\_increment not null primary key,  Fecha\_Pago date,  Precio int(6) unsigned not null,  ID\_Publicidad int(6) unsigned not null,  foreign key ID\_Publicidad references Publicidad(ID\_Publicidad)  );  ---  create table Tanda  (  Hora\_Inicio time not null primary key,  Hora\_Fin time not null  );  create table AparecePubli  (  ID\_Publicidad int(6) unsigned not null,  Hora\_Inicio time not null,  Fecha\_Inicio date not null,  Fecha\_Finalizacion date,  primary key(ID\_Publicidad,Hora\_Inicio),  foreign key ID\_Publicidad references Publicidad(ID\_Publicidad),  foreign key Hora\_Inicio references Tanda(Hora\_Inicio)  );  ---  create table Evento  (  ID\_Evento INT(6) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,  nombre VARCHAR(64) NOT NULL,  descripcion VARCHAR(256) NOT NULL  );  create table FechaEvento  (  ID\_Evento INT(6) unsigned,  fecha DATETIME NOT NULL,  FOREIGN KEY ID\_Evento REFERENCES Evento(ID\_Evento),  PRIMARY KEY(ID\_Evento, fecha)  );  create table EventoMuestraPubli  (  ID\_Evento int(6) unsigned not null,  ID\_Publicidad int(6) unsigned not null,  primary key(ID\_Evento, ID\_Publicidad),  foreign key ID\_Publicidad references Publicidad(ID\_Publicidad),  foreign key ID\_Evento references Evento(ID\_Evento)  );  create table EventoEsDeP  (  ID\_Programa int(6) unsigned not null,  ID\_Evento int(6) unsigned not null,  primary key(ID\_Programa,ID\_Evento),  foreign key ID\_Programa references Programa(ID\_Programa),  foreign key ID\_Evento references Evento(ID\_Evento)  );  create table Funcionario  (  ID\_Funcionario INT(6) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,  telefono INT(9) NOT NULL,  nombre VARCHAR(64) NOT NULL  );    create table Funcion  (  ID\_Funcion INT(6) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,  descripcion VARCHAR(256) NOT NULL  );  create table TrabajaComo  (  ID\_Funcionario INT(6) unsigned,  ID\_Funcion INT(6) unsigned,  FOREIGN KEY ID\_Funcionario REFERENCES Funcionario(ID\_Funcionario),  FOREIGN KEY ID\_Funcion REFERENCES Funcion(ID\_Funcion),  PRIMARY KEY(ID\_Funcionario, ID\_Funcion)  );  create table FunTrabaja  (  ID\_Funcionario INT(6) unsigned,  ID\_Funcion INT(6) unsigned,  ID\_Programa INT(6) unsigned,  fecha\_inicio DATE NOT NULL,  fecha\_finalizacion DATE,  FOREIGN KEY ID\_Funcionario REFERENCES Funcionario(ID\_Funcionario),  FOREIGN KEY ID\_Funcion REFERENCES Funcion(ID\_Funcion),  FOREIGN KEY ID\_Programa REFERENCES Programa(ID\_Programa),  PRIMARY KEY(ID\_Funcionario, ID\_Funcion, ID\_Programa)  );  create table Video  (  ID\_Video INT(6) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,  contenido VARCHAR(256) NOT NULL,  nombre VARCHAR(64) NOT NULL,  ID\_Serie INT(6) unsigned,  FOREIGN KEY ID\_Serie REFERENCES Serie(ID\_Serie),  fecha DATETIME  );    create table Serie  (  ID\_Serie INT(6) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,  fecha\_finalizacion DATE,  nombre VARCHAR(64) NOT NULL  );    create table VEsDeEvento  (  ID\_Evento INT(6) unsigned,  ID\_Video INT(6) unsigned,  FOREIGN KEY ID\_Evento REFERENCES Evento(ID\_Evento),  FOREIGN KEY ID\_Video REFERENCES Video(ID\_Video),  PRIMARY KEY(ID\_Evento, ID\_Video)  ); |

|  |
| --- |
| Creación de usuario |
| CREATE USER 'Admin'@'localhost' IDENTIFIED BY 'administrador';  GRANT USAGE ON \*.\* TO 'Admin'@'localhost';  GRANT ALL PRIVILEGES ON JVRPDATABASE.\* TO 'Admin'@'localhost';  FLUSH PRIVILEGES;  CREATE USER 'Invitado'@'localhost' IDENTIFIED BY 'invitado';  GRANT USAGE ON \*.\* TO 'Invitado'@'localhost';  GRANT select ON JVRPDATABASE.\* TO 'Invitado'@'localhost';  FLUSH PRIVILEGES; |

# Presentación del manual:

* Justificación del manual bilingüe.

Un manual bilingüe tiene ventajas ya que normalmente se lo crea en el idioma que el producto fue creado y en un idioma universal (ingles) para que todos los usuarios estén al mismo nivel de compresión del funcionamiento de dicho producto.

El usuario que tenga la posibilidad de acceder a un manual bilingüe ayuda a la compresión del programa sin que este sepa el idioma que originalmente el manual fue redactado. Además de ayudar a la compresión de cada función que el programa brinda en su idioma, sin tener la necesidad de estar traduciendo. También el manual bilingüe ayuda a una mejor comunicación y comprensión de dos usuarios que no tengan el mismo idioma y estén interesados en el programa.

* Formas de crear un manual bilingüe accesible.

Para la creación del manual bilingüe hay que realizar ciertos pasos ayudándonos con algunas herramientas que nos permitirán una mejor accesibilidad para el usuario.

Primero hay que identificar a que tipos de usuarios irá dirigido el manual, en el caso de ser un manual bilingüe hay que tener en cuenta los dos idiomas en los cuales será redactado, en nuestro caso español e inglés.

Al momento de la identificación de los usuarios que tendrán conocimientos sobre el manual, hay que tener en cuenta la capacitación que estos tienen, normalmente es mejor crear el manual a un nivel de usuario estándar, ya que puede haber un usuario que no tenga una gran capacitación.

El formato y el estilo del manual debe ser apropiado, debe tener un formato sencillo pero coherente, el estilo de la escritura puede ser formal (si está dirigido para usuarios ya con conocimientos) o semi-formal. El seguimiento de la escritura debe ser fácil para que el usuario pueda seguir y encontrar con facilidad la información necesaria, además de implementar imágenes de guía. También se diferencia entre los títulos, subtítulos, advertencias, cuidados, etc.

El estilo y el formato del manual son elegidos por los miembros del equipo que se pondrán de acuerdo luego reconocer el tipo de usuario al cual este manual será dirigido. A raíz de eso se comenzará con la creación del mismo, preferentemente se sugiere ir creando el manual en un idioma para luego traducirlo.

Debido a que el destinatario del manual poseería conocimientos previos sobre el uso de dispositivos y herramientas en relación a la informática y la comunicación, nuestro manual omitirá explicaciones introductorias al uso de sistemas. Esto nos lleva a describir exclusivamente la organización, acceso, permisos y posibilidades implementadas dentro del software.

Primero se comienza con identificación de las partes que tendrá el mismo, especificando que irá primero antes de la inicialización del programa, que debe hacer el usuario para que este inicie correctamente, adjuntando imágenes de guía y el listado de pasos. Luego se deberá especificar que hará el usuario luego de haberlo hecho, en nuestro caso que es lo primero que ve el usuario al ingresar al programa, explicando que es lo que el usuario está viendo y que es lo que debe hacer. El manual irá especificando el uso de cada interfaz que el usuario está visualizando, con sus respectivos títulos, sus imágenes de guía, pasos a seguir, etc.

Luego de la realización de la especificación del programa y su funcionamiento, se deberá realizar un índice, para una mejor ayuda al momento de buscar alguna información concreta que el usuario desee, sin que este tenga que estar leyendo todo el manual para encontrarla.

Al tener finalizado el manual en un idioma se comienza con la traducción de este, teniendo en cuenta la estructura y organización, haciendo el mismo procedimiento que se realizó al comienzo.

Para poder finalizar la realización del manual bilingüe se exporta el mismo en los formatos pdf, docx y txt (sin imágenes en este último formato), además para una mejor distribución se lo imprime. En conjunto poseería un archivo del manual en formato de audio.

Manual:

* 1. Primera ventana:



El primer formulario que se observa requiere el ingreso del usuario en el primer campo en blanco, la contraseña en el segundo.

Luego de esto y si los datos son válidos se podría abrir la ventana de configuración o entrar al sistema.

* 1. Ventana de configuración



Lo primero que se notará es que posee varios campos de texto para ingresar los datos de la conexión, con el correspondiente tipo de dato que se requiere. Posee la opción de restablecer para volver a los valores por defecto, o guardar los datos ingresados en el momento.

También posee en el panel una pestaña para administrar los usuarios (solo está disponible si el usuario tiene dicho permiso).

* 1. Pestaña de usuario



En la pestaña de usuario se puede observar una sección para la edición y otra para la creación de usuarios.

En la sección de creación hay un campo para el nombre y la contraseña para el usuario a crear, además de dos botones uno para crear y otro para limpiar.

En la sección de edición se observa una tabla con el id y el nombre de usuario, y al seleccionar alguno este se cargará en la sección de edición, pudiendo modificar el nombre y/o la contraseña. Siguiente a esto se puede aplicar los cambios a los usuarios o eliminar dicho usuario.

Código:

Módulos:

ModCodificador:

Imports System.Text

Imports System.Security.Cryptography

'Se utiliza para encriptar y desencriptar datos de tipo string

'Usa una key y un vector de incializacion, que llamaremos key maestra

'Por mas información de esta ultima se puede buscar en:

'https://es.wikipedia.org/wiki/Vector\_de\_inicializaci%C3%B3n

Module ModCodificador

Private KeyMaestra As String = "rpaSPvIvVLlrcmtzPU9/c67Gkj7yL1S5"

Private Key As String = "12345678"

Private IV() As Byte

Private EncryptionKey() As Byte

Private buffer() As Byte

Private des As TripleDESCryptoServiceProvider

Public Sub EKey(ByVal NKey As String)

If Not String.IsNullOrEmpty(NKey) Then

Key = NKey

End If

End Sub

Public Sub Actualizar()

IV = ASCIIEncoding.ASCII.GetBytes(Key) 'La clave debe ser de 8 caracteres

EncryptionKey = Convert.FromBase64String(KeyMaestra) 'No se puede alterar la cantidad de caracteres pero si la clave

des = New TripleDESCryptoServiceProvider

des.Key = EncryptionKey

des.IV = IV

End Sub

Public Sub EKeyMaestra(ByVal NKeyMaestra As String)

If Not String.IsNullOrEmpty(NKeyMaestra) Then

KeyMaestra = NKeyMaestra

End If

End Sub

Public Function GKey() As String

Return Key

End Function

Public Function GKeyMaestra() As String

Return KeyMaestra

End Function

Public Function Encriptar(ByVal Texto As String) As String

'convierte el texto a bytes utilizando el formato utf8

buffer = Encoding.UTF8.GetBytes(Texto)

'Crea un objeto de encriptacion, luego crea un valor hash con el buffer y utiliza todo su largo, comenzando en el byte 0

Return Convert.ToBase64String(des.CreateEncryptor().TransformFinalBlock(buffer, 0, buffer.Length()))

End Function

Public Function Desencriptar(ByVal Texto As String) As String

'convierte el texto encriptado a bytes

buffer = Convert.FromBase64String(Texto)

'desencripta el texto(ahora en array de bytes) y utiliza el formato utf8

Return Encoding.UTF8.GetString(des.CreateDecryptor().TransformFinalBlock(buffer, 0, buffer.Length()))

End Function

End Module

ModConector:

Imports MySql.Data.MySqlClient

Imports System.Data

Imports MySql.Data

Module ModConector

Private Debug As Boolean = True

Private conn As New MySqlConnection

Private connStr As String

Private Adress, User, Database, Port, Pass As String

Private UsuarioID As Integer

Private Usuario, Password As String

Private objCmd As New MySqlCommand

Private ds As New DataSet

Private dt As New DataTable

Private sqladapter As MySqlDataAdapter

#Region "GetDirection"

Public Function GDT(ByVal Nombre As String) As DataTable

Return ds.Tables(Nombre)

End Function

Public Function GUsuarioID() As Integer

Return UsuarioID

End Function

Public Sub BorrarUsuario()

Usuario = ""

Password = ""

UsuarioID = Nothing

End Sub

Public Function GDebug() As Boolean

Return Debug

End Function

Public Function GAdress() As String

Return Adress

End Function

Public Function GUser() As String

Return User

End Function

Public Function GDatabase() As String

Return Database

End Function

Public Function GPort() As String

Return Port

End Function

Public Function GPass() As String

Return Pass

End Function

#End Region

#Region "Constructor"

Public Sub Crear(ByVal RAdress As String, ByVal RPort As String, ByVal RDatabase As String, ByVal RUser As String, ByVal RPass As String)

If Not String.IsNullOrEmpty(Adress) Then

Adress = RAdress

Else

Adress = "localhost"

End If

If Not String.IsNullOrEmpty(Adress) Then

User = RUser

Else

User = "root"

End If

If Not String.IsNullOrEmpty(Adress) Then

Database = RDatabase

Else

Database = "JVRPDB"

End If

If Not String.IsNullOrEmpty(Adress) Then

Port = RPort

Else

Port = "3306"

End If

If Not String.IsNullOrEmpty(Adress) Then

Pass = RPass

Else

Pass = "root"

End If

End Sub

Public Sub EAdress(ByVal NAdress As String)

Adress = NAdress

End Sub

Public Sub EPort(ByVal NPort As String)

Port = NPort

End Sub

Public Sub EUser(ByVal NUser As String)

User = NUser

End Sub

Public Sub EPass(ByVal NPass As String)

Pass = NPass

End Sub

Public Sub EDatabase(ByVal NDatabase As String)

Database = NDatabase

End Sub

#End Region

#Region "Conectar"

Public Sub Inicio()

Try

connStr = "DataSource=" + Adress + "; Port=" + Port + "; Database=" + Database + "; Uid=" + User + "; Pwd=" + Pass + "; CharSet=utf8"

'connStr = "Server=" + Adress + "; Database=" + Database + "; Uid=" + User + "; Pwd=" + Pass + "; CharSet=utf8mb4"

'connStr = "Server=" + Adress + "; Database=" + Database + "; Uid=" + User + "; Pwd=root; CharSet=utf8mb4"

conn = New MySqlConnection(connStr)

If Not conn.Ping Then

conn.Open()

End If

Catch ex As Exception

MessageBox.Show(ex.ToString())

End Try

End Sub

Public Sub desconectar()

conn.Close()

End Sub

Public Function RConexion() As MySqlConnection

Return conn

End Function

#End Region

#Region "Interpretar"

Public Function ESQL(ByVal sql As String) As Boolean

Try

objCmd = New MySqlCommand(sql, conn)

objCmd.Prepare()

objCmd.ExecuteNonQuery()

Catch ex As Exception

MessageBox.Show(ex.ToString)

Return False

End Try

Return True

End Function

Public Function ESQLSelect(ByVal sql As String) As DataTable

Dim dt As New DataTable

Try

objCmd = New MySqlCommand(sql, conn)

objCmd.Prepare()

sqladapter = New MySqlDataAdapter(objCmd)

sqladapter.Fill(dt)

ds.Tables.Add(dt)

Catch ex As Exception

MessageBox.Show(ex.ToString)

Return Nothing

End Try

Return dt

End Function

Public Function ESQLSelect(ByVal objCmd As MySqlCommand, ByVal guardar As Boolean) As DataTable

dt = New DataTable

Try

sqladapter = New MySqlDataAdapter(objCmd)

sqladapter.Fill(dt)

If guardar Then

If ds.Tables.Contains(dt.TableName) Then

ds.Tables.Remove(dt.TableName)

End If

ds.Tables.Add(dt)

End If

Catch ex As Exception

MessageBox.Show(ex.ToString)

Return Nothing

End Try

Return dt

End Function

#End Region

#Region "Comandos"

Public Sub BSQL(ByVal nTabla As String, ByVal Condition As String)

Dim borrar As String = "DELETE FROM " + nTabla + " WHERE " + Condition

ESQL(borrar)

End Sub

Public Sub ISQL(ByVal nTabla As String, ByVal Column As String, ByVal Data As String)

Dim insert As String = "Insert into " + nTabla + " ( " + Column + " ) values (" + Data + " )"

ESQL(insert)

End Sub

Public Function SSQL(ByVal nColumn As String, ByVal nTabla As String, ByVal Condition As String) As DataTable

Dim selector As String = "select " + nColumn + " FROM " + nTabla + " WHERE " + Condition

If ds.Tables.Contains(nTabla) Then

dt = New DataTable

dt.Columns.Add(ds.Tables(nTabla).Columns(nColumn))

Return dt

Else

Return ESQLSelect(selector)

End If

End Function

Public Sub USQL(ByVal nTabla As String, ByVal Orden As String, ByVal Condition As String)

Dim update As String = "update " + nTabla + " set " + Orden + " WHERE " + Condition

ESQL(update)

End Sub

#End Region

#Region "Usuarios"

Public Function AUsuarios(ByVal actualizar As Boolean) As DataTable

If ds.Tables.Contains("usuarios") And Not actualizar Then

dt = New DataTable()

dt.Columns.Add(ds.Tables("usuarios").Columns("Nombre Usuarios"))

Return dt

Else

Try

objCmd = New MySqlCommand("SELECT id\_usuario as 'ID', nombre as 'Nombre Usuarios' FROM usuarios", conn)

objCmd.Prepare()

Dim dt As DataTable = ESQLSelect(objCmd, True)

If Not IsNothing(dt) Then

If dt.Rows.Count = 0 Then

MessageBox.Show("No hay usuarios")

Else

Return dt

End If

End If

Catch e As Exception

MessageBox.Show(e.ToString)

End Try

Return Nothing

End If

End Function

Public Function BUsuario(ByVal nombre As String, ByVal contraseña As String) As Boolean

Try

objCmd = New MySqlCommand("SELECT id\_usuario FROM usuarios as User WHERE nombre = @nombre AND contrasena = AES\_ENCRYPT(@contrasena,sha2(@key,256))", conn)

objCmd.Parameters.Add("@nombre", MySqlDbType.VarChar).Value = nombre

objCmd.Parameters.Add("@contrasena", MySqlDbType.VarChar).Value = contraseña

objCmd.Parameters.Add("@key", MySqlDbType.VarChar).Value = ModCodificador.GKey

objCmd.Prepare()

Dim dt As DataTable = ESQLSelect(objCmd, False)

If Not IsNothing(dt) Then

If dt.Rows.Count = 0 Then

MessageBox.Show("Contraseña o Usuario Incorrecto")

Else

Usuario = nombre

Password = contraseña

UsuarioID = Integer.Parse(dt.Rows(0)("id\_usuario"))

Return True

End If

End If

Catch e As Exception

MessageBox.Show(e.ToString)

End Try

Return False

End Function

Public Sub IUsuario(ByVal nombre As String, ByVal contraseña As String)

ISQL("usuarios", "nombre , contrasena", "'" + nombre + "', AES\_ENCRYPT('" + contraseña + "',sha2('" + ModCodificador.GKey + "',256))")

End Sub

#End Region

End Module

ModuloIniciador:

' El modulo trabaja como union y acceso entre los formularios

Module ModuloInicializador

Public Sub Configuracion()

Dim frmConfig As New frmConfiguracion

' Muestra el formulario de configuracion

frmConfig.ShowDialog()

End Sub

Public Sub Principal()

Dim frmPrin As New frmPrincipal

' Muestra el formulario de configuracion

frmPrin.Show()

End Sub

End Module

ModUser:

Imports System

Imports System.IO

Imports System.Collections

Module ModUser

':::Ruta donde crearemos nuestro archivo txt

Private ruta As String = "..\User\"

':::Nombre del archivo

Private archivo As String = "User.txt"

Private Barchivo As String = "Origin.txt"

Private karchivo As String = "Key.txt"

'Establece la key y una verificacion

Public Sub Inicio()

If Not Directory.Exists(ruta) Then

Directory.CreateDirectory(ruta)

End If

Dim leer As New StreamReader(ruta & karchivo)

If File.Exists(ruta & karchivo) Then

ModCodificador.EKey(leer.ReadLine())

ModCodificador.EKeyMaestra(leer.ReadLine())

Else

If File.Exists(ruta & archivo) Then

File.Delete(ruta & archivo)

End If

If File.Exists(ruta & Barchivo) Then

File.Delete(ruta & Barchivo)

End If

File.Create(ruta & karchivo)

Dim escribir As New StreamWriter(ruta & karchivo, False)

escribir.WriteLine(ModCodificador.GKey())

escribir.WriteLine(ModCodificador.GKeyMaestra())

escribir.Close()

End If

leer.Close()

Verify()

ModCodificador.Actualizar()

LeeDatos()

End Sub

Public Sub Verify()

If Not File.Exists(ruta & archivo) Then

If Not File.Exists(ruta & Barchivo) Then

File.Create(ruta & Barchivo)

Else

File.Copy(ruta & Barchivo, ruta & archivo)

End If

End If

End Sub

Public Sub Borrar()

If File.Exists(ruta & archivo) Then

File.Delete(ruta & archivo)

End If

End Sub

Public Sub LeeDatos()

If File.Exists(ruta & archivo) Then

Dim leer As New StreamReader(ruta & archivo)

Dim Datos(4) As String

Dim n As Byte = 0

While leer.Peek <> -1

'Leemos cada linea del archivo TXT

Dim linea As String = leer.ReadLine()

'Agregramos los datos

Datos(n) = ModCodificador.Desencriptar(linea)

n += 1

End While

ModConector.Crear(Datos(0), Datos(1), Datos(2), Datos(3), Datos(4))

leer.Close()

End If

End Sub

Public Sub Guardar(ByVal Desarrollo As Boolean)

Dim archivos As String

If Desarrollo Then

archivos = Barchivo

Else

archivos = archivo

End If

Dim escribir As New StreamWriter(ruta & Archivos, False)

escribir.WriteLine(ModCodificador.Encriptar(ModConector.GAdress()))

escribir.WriteLine(ModCodificador.Encriptar(ModConector.GPort()))

escribir.WriteLine(ModCodificador.Encriptar(ModConector.GDatabase()))

escribir.WriteLine(ModCodificador.Encriptar(ModConector.GUser()))

escribir.WriteLine(ModCodificador.Encriptar(ModConector.GPass()))

escribir.Close()

End Sub

End Module

Sección Usuario:

Configuración:

Public Class frmConfiguracion

Private UserID As Integer = Nothing

Private UsuarioDatos As DataTable

Public dt As DataTable

Private Sub btnRestablecer\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRestablecer.Click

ModUser.Borrar()

ModUser.Inicio()

End Sub

Private Sub btnGuardar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGuardar.Click

If (txtIp.Text <> "") Then

ModConector.EAdress(txtIp.Text)

End If

If (txtPuerto.Text <> "") Then

ModConector.EPort(txtPuerto.Text)

End If

If (txtUsuario.Text <> "") Then

ModConector.EUser(txtUsuario.Text)

End If

ModConector.EPass(txtContraseña.Text)

If (txtBD.Text <> "") Then

ModConector.EDatabase(txtBD.Text)

End If

ModUser.Guardar(False)

ModUser.LeeDatos()

End Sub

Private Sub DebugCrear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DebugCrear.Click

If (txtIp.Text <> "") Then

ModConector.EAdress(txtIp.Text)

End If

If (txtPuerto.Text <> "") Then

ModConector.EPort(txtPuerto.Text)

End If

If (txtUsuario.Text <> "") Then

ModConector.EUser(txtUsuario.Text)

End If

ModConector.EPass(txtContraseña.Text)

If (txtBD.Text <> "") Then

ModConector.EDatabase(txtBD.Text)

End If

ModUser.Guardar(True)

ModUser.LeeDatos()

Me.Dispose()

End Sub

Private Sub DebugCrearUsuario\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

ModConector.IUsuario(txtUsuario.Text, txtContraseña.Text)

End Sub

Private Sub frmConfiguracion\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

ModUser.Inicio()

ModConector.Inicio()

End Sub

Private Sub frmConfiguracion\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

If ModConector.GDebug Then

DebugCrear.Visible = True

DebugCrear.Enabled = True

End If

Control.CheckForIllegalCrossThreadCalls = False

End Sub

Private Sub TabPage2\_Enter(ByVal sender As Object, ByVal e As System.EventArgs) Handles TabPage2.Enter

'ActualizarUsuarios(False)

BW.RunWorkerAsync(False)

LimpiarEditar()

limpiar()

End Sub

#Region "Limpiadores"

Private Sub limpiar()

txtNombre.Text = ""

txtContrasena.Text = ""

dgvNombreUsuario.ClearSelection()

End Sub

Private Sub LimpiarUsuario\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LimpiarUsuario.Click

limpiar()

End Sub

Private Sub LimpiarEditar()

txtENombre.Text = ""

txtEContrasena.Text = ""

UserID = Nothing

End Sub

Private Sub UAplicar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UAplicar.Click

If Not IsNothing(UserID) Then

USQL("usuarios", "nombre ='" + txtENombre.Text + "',contrasena = AES\_ENCRYPT('" + txtEContrasena.Text() + "',sha2('" + ModCodificador.GKey + "',256))", "id\_usuario ='" + UserID.ToString() + "'")

'ActualizarUsuarios(False)

BW.RunWorkerAsync(False)

If UserID = ModConector.GUsuarioID Then

ModConector.BorrarUsuario()

Me.Dispose()

End If

End If

LimpiarEditar()

End Sub

#End Region

#Region "Modificar Usuarios"

Private Sub UBorrar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UBorrar.Click

If Not IsNothing(UserID) Then

BSQL("usuarios", "id\_usuario ='" + UserID.ToString + "'")

'ActualizarUsuarios(False)

BW.RunWorkerAsync(False)

If UserID = ModConector.GUsuarioID Then

ModConector.BorrarUsuario()

Me.Dispose()

End If

End If

LimpiarEditar()

End Sub

Private Sub ActualizarUsuarios()

If Not IsNothing(dt) Then

Me.dgvNombreUsuario.DataSource = dt

Else

MessageBox.Show("No se cargo correctamente.")

End If

End Sub

Private Sub CrearUsuario\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CrearUsuario.Click

ModConector.IUsuario(txtNombre.Text, txtContrasena.Text)

'ActualizarUsuarios(True)

BW.RunWorkerAsync(True)

limpiar()

End Sub

#End Region

Private Sub dgvNombreUsuario\_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles dgvNombreUsuario.CellMouseClick

Try

LimpiarEditar()

txtENombre.Text = dgvNombreUsuario.CurrentRow.Cells("Nombre Usuarios").Value.ToString

UserID = dgvNombreUsuario.CurrentRow.Cells("ID").Value

Catch es As Exception

End Try

End Sub

Private Sub BW\_DoWork(ByVal sender As Object, ByVal e As System.ComponentModel.DoWorkEventArgs) Handles BW.DoWork

dt = ModConector.AUsuarios(e.Argument)

End Sub

Private Sub BW\_RunWorkerCompleted(ByVal sender As Object, ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) Handles BW.RunWorkerCompleted

ActualizarUsuarios()

End Sub

Private Sub GroupBox1\_Enter(sender As Object, e As EventArgs) Handles GroupBox1.Enter

End Sub

End Class

Formulario usuario:

Public Class frmLUsuario

Private Sub btnOpciones\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOpciones.Click

If BUsuario(txtUsuario.Text, txtPass.Text) Or ModConector.GDebug Then

ModuloInicializador.Configuracion()

End If

End Sub

Private Sub btnSalir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

End

ModConector.desconectar()

End Sub

Private Sub Limpiar()

txtUsuario.Text = ""

txtPass.Text = ""

End Sub

Private Sub btnEntrar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEntrar.Click

If ModConector.BUsuario(txtUsuario.Text, txtPass.Text) Then

MessageBox.Show("Bienvenido")

ModuloInicializador.Principal()

Me.SetVisibleCore(False)

End If

End Sub

Private Sub frmLUsuario\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

ModConector.desconectar()

End Sub

Private Sub frmLUsuario\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

ModUser.Inicio()

ModConector.Inicio()

Catch m As Exception

MessageBox.Show(e.ToString)

End Try

End Sub

Private Sub lblPass\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles lblPass.Click

End Sub