|  |
| --- |
| Universidad Técnica Nacional |
| Carrera: T.I.  Curso ISW-411 : Programación III - C #  Prof. : MRT. Anthony Morera V.  Periodo : III - 2016 |

|  |
| --- |
| Proyecto Programado |

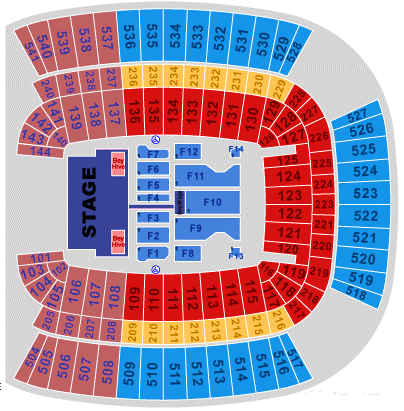
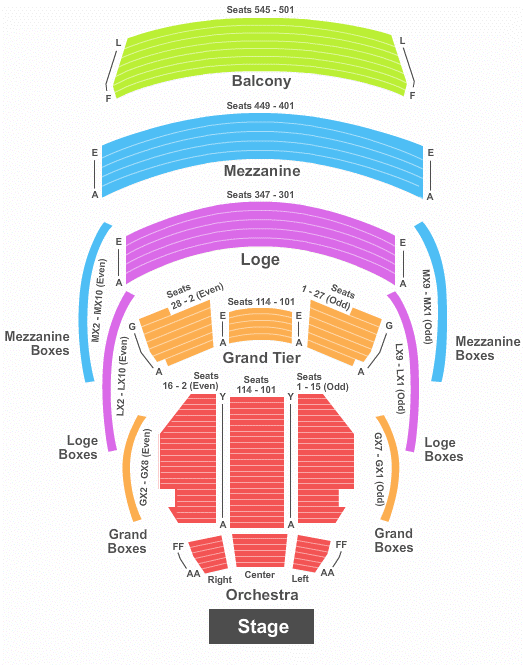
**Valor: 20% del Nota Final**

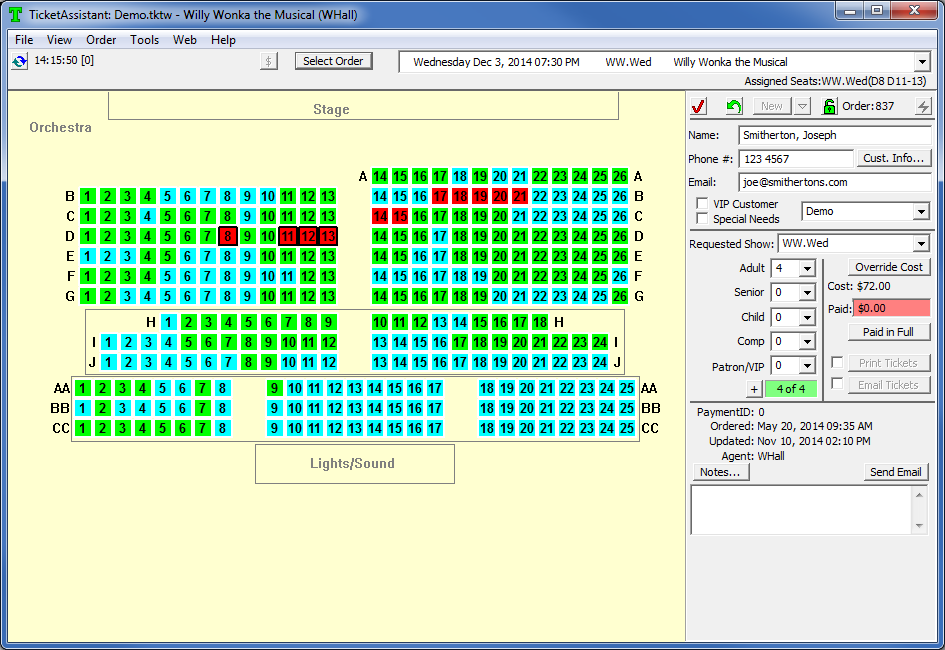
|  |
| --- |
| A- Descripción del Proyecto |

Se le ha asignado el Análisis, Diseño, Arquitectura y Desarrollo de un Sistema que permita administrar las ventas de entradas a eventos como Conciertos, Partidos, Actividades, Cine, Teatro entre otros.

El sistema debe ser innovador, configurable y utilizar al máximo la interfaz gráfica.





|  |
| --- |
| B- Requerimientos funcionales |

1. Nombre del Sistema y un logo alusivo.
2. Control de:
   1. Eventos.
   2. Trabajadores.
   3. Clientes
   4. Facturación y emisión de entradas.
   5. Reportes varios.
   6. Seguridad
3. Los nombres utilizados deben ser referentes a la temática.
4. La interfaz gráfica orientada a la utilización de dispositivos táctiles para que sea fácil de utilizar por los usuarios.

|  |
| --- |
| C- Menú del Sistema Solicitado |
| A continuación se citan los ítems mínimos recomendados/requeridos del menú del Sistema principal. |
| 1. Mantenimientos |

1. **Empleados**

Incluir los datos del empleado nombre, apellidos, teléfonos, correo, dirección identificación y la **fotografía**.

1. **Tipos de Eventos.**

Código del Tipo, descripción (Conciertos, Evento Religioso, Partidos, Motocross entre otros)

1. **Configurador de Eventos**

Código de Evento, tipo de Evento, descripción, imagen holográfica o logo del evento, precio del boleto por zona, costos de administrativos, fecha y hora del evento, tipos de localizaciones, cantidad por localización, estado del evento(activo|inactivo), cantidad de entradas que puede comprar un cliente.

1. **Clientes**

Código de Cliente, identificación, nombre completo, fecha de nacimiento, sexo, correo electrónico (**obligatorio**), nacionalidad y cualquier otro que usted considere importante.

|  |
| --- |
| 2. Procesos |

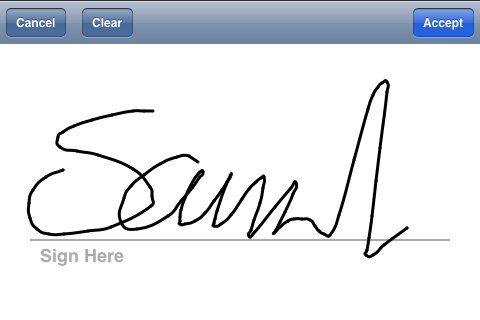
* 1. **Facturación**

El usuario debe acceder al sistema y seleccionar el tipo evento, validar si el evento está accesible, un evento puede estar cancelado, si tiene espacios disponibles, si no ha pasado la fecha del evento, se le pedirán los datos al cliente, si no existe se agrega en la base de clientes. Es importante que solo se permiten pagos por tarjetas de crédito American Express, Visa, Master Card y Discover.



Tome en cuenta que la cantidad de entradas que puede comprar un mismo cliente por evento, “puede” estar limitada.

La firma del boucher se hace de forma electrónica en la misma factura y se almacena en la base de datos. **Desarrolle y proponga una solución a este requerimiento.**



Una vez que se pagó el o los tiquetes de las entradas la transacción se dará como paga, luego el sistema enviará dos correos.

* + 1. Factura electrónica al correo con el detalle de la compra y la información más relevante. La factura mostrar todo el detalle como cliente, fecha, detalle, impuestos y su número de factura tanto visual como en QuickResponse[[1]](#footnote-1).



* + 1. Entrada electrónica para ser impresa o leída desde el móvil y que contendrá el código QuickResponse con el número de entrada y la imagen holográfica o el logo del evento.



Antes de enviar las facturas y las entradas, el sistema preguntará al usuario si desea enviarlas al correo que es lo óptimo.

El sistema siempre pre visualizará la factura y las entradas también permitirá salvarlas en disco. Se debe contemplar que no haya internet o que no se pueda enviar por ello el sistema debe preguntar si se desea enviar la información al correo del cliente y dar la posibilidad de salvarla en disco.

Este procedimiento debe venir probado y ejecutado, se recomienda tener una cuenta de prueba con los correos enviados.

|  |
| --- |
| 3. Reportes |

1. Reporte de Ventas por evento **diarias por rangos de fechas**.
2. Reporte de un Evento, donde se visualice los espacios libres y ocupados por **evento y rango de fecha**.

|  |
| --- |
| 4. Administración |

a- Creación de usuarios del Sistema: Todas las calidades del trabajador e incluye la fotografía.

b- Asignar perfil: Asignación de los perfiles.

Hay 3 tipos de **perfiles**

1. **ADMINISTRADOR** (todas las opciones del menú)
2. **PROCESOS** (Ejecutar los procesos)
3. **REPORTEADOR** (Sólo Reportes)

Sólo el usuario **Administrador** puede acceder a esta opción del menú.

Nota: El Administrador tendrá derechos para crear usuarios a nivel de base de datos.

|  |
| --- |
| D- Requerimientos Técnicos del Sistema |

1. Lenguaje C# 4.X con el Framework 4.5. en Visual Studio 2013 Winforms con todos los objetos gráficos disponibles.
2. Debe crear la base de datos en SqlServer 2012. El modelo debe estar normalizado hasta 3FNBC
3. La información DEBEN SER REAL, se recomienda realizar una investigación sobre cada tema.
4. Debe utilizar el Paradigma Orientado a Objetos (Abstracción, Encapsulamiento, Modularidad, Jerarquía) y utilizar los patrones básicos GRASP, además de otros Patrones vistos en clase (Facade, Factory, CRUD, Singleton, Active Record, N-Tiers, Data Transfer Object, Proxy, Observador etc )
5. El proyecto debe ser programado bajo el patrón de Arquitectura de "N-Capas", utilizando el Paradigma de Orientación a Objetos. Se solicita ordenar la solución del Proyecto en diferentes carpetas por ejemplo.
   * + 1. **Layers**
          1. **DAL**
          2. **BLL**
          3. **UI (Winforms)**
          4. **Entities**

**DTO**

**Business Objects**

**Factories**

* + - * 1. **Interface**
      1. **Enumeraciones**
      2. **Util (Clases de utilitarios como por ejemplo control de trazas de seguimientos)**

1. Todos los Mantenimientos (Registros) deben ser realizados con Procedimientos Almacenados utilizando el Patrón de CRUD.
2. El sistema debe poseer seguridad por perfiles. No se podrá utilizar el usuario **sa** para realizar transacciones.
3. El proyecto debe venir documentado con Doxigen[[2]](#footnote-2)
4. La entrega del Proyecto debe ser en un CD debidamente etiquetado con una carpeta llamada Proyecto con los fuentes, otra carpeta llamada BaseDatos y otra Documentación (archivo chm)
5. Utilizar estándares de programación (botones, ventanas, nombres de objetos entre otros) y comentarios en el código.
6. Debe realizar trazas de eventos en logs utilizando Log4NET[[3]](#footnote-3).

|  |
| --- |
| E- Evaluación |

**La defensa consiste en traer varios casos listos donde involucre toda la funcionalidad del Sistema. Se recomienda traer 5 eventos, 2 cerrados y 3 en proceso.**

|  |  |  |
| --- | --- | --- |
| Puntos a Evaluar | Puntos | Obtenidos |
| Registros Ejemplo | 10 pts |  |
| Diseño de la Arquitectura del Software | 15 pts |  |
| Mantenimientos | 20 pts |  |
| Procesos | 40 pts |  |
| Reportes | 10 pts |  |
| Defensa del Proyecto |  |  |
|  | 5 pts |  |
| Total  100 pts | 100 pts |  |
| Porcentaje | 20 % |  |

|  |
| --- |
| Nota: |
| 1. El proyecto es estrictamente individual. |
| 1. En caso de no defender el proyecto correctamente, la nota quedará a criterio del Profesor. |
| 1. De encontrarse Proyectos iguales/similares serán trasladados a la Coordinación de Carrera. |

1. https://es.wikipedia.org/wiki/C%C3%B3digo\_QR [↑](#footnote-ref-1)
2. http://www.stack.nl/~dimitri/doxygen/ [↑](#footnote-ref-2)
3. https://logging.apache.org/log4net/ [↑](#footnote-ref-3)