



## PROYECTO FINAL DE SEGUNDO AÑO

CARRERA DE ANALISTA DE SISTEMAS  
ANALISTA PROGRAMADOR WEB .NET  
2024

### Generalidades

- La entrega final se hará durante la última clase de tutoría. Durante las clases de tutorías deberán realizarse entregas parciales de acuerdo a lo establecido en la sección Tutorías & Entregas.
- Se deberá realizar la entrega mediante correo electrónico, del software junto con toda la documentación exigida en la sección **Requerimientos de Entrega**:
  - **Asunto:** Entrega Proyecto Final ANP 2024
  - **Destinatarios:** segundoanalista@bios.edu.uy y tutor asignado (OBLIGATORIO)
  - **Compartido por Drive:** documentación y solución completa
  - **Cuerpo:** Nombre y Cedula de los integrantes del grupo que realiza la entrega
- Se deberán formar grupos de 1 a 3 personas, los cuales deberán inscribirse en coordinación académica desde el día 25 de noviembre hasta el día 06 de diciembre. **No se aceptarán inscripciones fuera de fecha.** Luego de esto, se **publicará** la asignación de tutores, así como de horarios. **La inscripción es únicamente por vía mail (ibalciunas@bios.edu.uy), con el siguiente formato:**
  - a) Asunto: *Inscripción a Proyecto Final Segundo Año*
  - b) Contenido:
    - a. Cedula y Nombre de ***todos*** los integrantes del grupo que se presenta
    - b. Franja Horaria a la que se concurre a clase
- La asistencia a la última tutoría es obligatoria para **todos** los miembros del grupo ya que se realizará la defensa en máquina del proyecto.



## Requerimientos de Implementación

- Implementación completa del sistema con tecnologías .NET en lenguaje C#. Obligatorio la entrega en **.Net Framework 4.5.2 o 4.6**.
- La información deberá almacenarse obligatoriamente en una base de datos **SQL Server**.
- El script de la base de datos debe generarse **manualmente**, sin la ayuda de un asistente. Deberá contener el **Esquema de creación de la base de datos**, **Stored Procedures** necesarios para realizar todas las tareas solicitadas, **Creación de usuarios y permisos** solicitados y necesarios, e **Inserción** de datos de prueba. Las restricciones sobre datos, deberán realizarse en la propia estructura de la tabla (uso de los modificadores **unique, check, default**).
- Los **Estados de las Ventas** en esta versión no tendrán mantenimiento a través del sistema, se deben generar a nivel de base de datos. Se deberán tener estos 4 estados: 1- armado // 2 – envío // 3 – entregado // 4 – devuelto. Los **Empelados** tampoco tendrán mantenimiento a través del sistema, se deben generar a nivel de base de datos.
- Las eliminaciones a nivel de la base de datos deberán ser físicas o lógicas en función de las posibilidades. Es decir, si un elemento a eliminar no tiene elementos dependientes dentro de los registros de la base de datos, o estos se pueden eliminar, se elimina físicamente; de lo contrario se hará una eliminación lógica. **Obligatorio de implementar**
- Para el desarrollo de la Aplicación Web MVC, se debe utilizar la arquitectura en 3 capas vista en el curso, mediante la utilización de bibliotecas de clases. Se deberán crear los proyectos correspondientes a las 3 bibliotecas y luego referenciarlas desde la aplicación. **Obligatorio de implementar**
- Para el desarrollo del Sitio Web ASP, se deberá utilizar Entity Framework DBF. Para ello se deberá generar una biblioteca para el modelo de Entity Framework y luego referenciarla desde el sitio. **Obligatorio de implementar**
- **Obligatorio** el uso de clases para la comunicación de datos entre componentes (tanto para invocación como respuesta).
- Los componentes de Lógica y Persistencia deberán ser generados en base a los patrones de **Fábrica y Singleton** vistos en clase. No deberá implementarse ningún otro patrón.
- Siempre se deberá de trabajar en forma conectada (ADO.NET) para el manejo de la información de la base de datos en el caso de la capa de **Persistencia**.
- Los componentes deberán lanzar excepciones en caso de error. No se contempla otra forma de comunicación de errores entre componentes

## Requerimientos de Entrega

- Modelo Conceptual.
- MER
- Diagrama de Clases completo de la Arquitectura en capas (incluye a todos los componentes, clases, interfaces y relaciones entre ellos)
- Solución completa del Software (2 proyectos de UI – 4 bibliotecas de clases)
- Script de la base de datos

*Nota: todos los diagramas deberán ser generados con una herramienta para lenguaje UML. Estos deberán ser entregados en forma digital: una copia del archivo original del diagrama y una copia en formato PDF o JPG*



## Idea General

Se desea realizar el desarrollo de un sistema de ventas para una farmacia (empresa que gestiona la venta de artículos farmacéuticos y el seguimiento de las entregas)

## Descripción de la Realidad

Los usuarios empleados (que gestionan los artículos, los clientes y las ventas) deberán tener usuario de logueo (**único** en el Sistema), contraseña y nombre

Los conceptos que forman parte de la realidad se describen en cada punto en donde se deben usar, para generar las funcionalidades del sistema propuesto

## Arquitectura Solicitada

- Para el acceso público, se deberá generar un Sitio Web ASP.Net, que trabajará con Entity Framework, Modelo DataBase First.
- Para el acceso de los empleados, se deberá generar una Aplicación Web MVC, que trabajará con 3 capas (Entidades, Lógica, Persistencia).
- La única base de datos estará instalada en un servidor de datos.
- Ambas UI Web deberán estar instaladas en un servidor IIS



## Funcionalidades Mínimas del Sitio Web ASP .Net

**Formulario Web:** Principal (formulario por defecto del sitio)

**Actor Participante:** público

**Resumen:**

- a) En este formulario se desplegarán por defecto en un control grilla todos los artículos que ofrece a la venta la empresa (código, nombre, fecha vencimiento), que **NO** estén vencidos. Deberán estar ordenados alfabéticamente por nombre. Además, dicha grilla deberá estar paginada
- b) Filtros:
  - 1. Se debe contar con un buscador que permita filtrar los artículos por su nombre.
  - 2. También se deberá contar con una lista desplegable de categorías de artículos, para poder filtrar, por categoría los artículos visualizados (es decir que, si la grilla ya tiene filtro por nombre, el filtro por categoría deberá aplicarse a esos artículos desplegados).
- c) Selección de un artículo: en ese caso se deberá desplegar toda la información completa de dicho artículo, incluyendo categoría y cantidad de ventas que tiene
- d) ***Uso obligatorio de Linq para obtener datos, ordenar y realizar filtros***



## Funcionalidades Mínimas de la Aplicación Web MVC

### Logueo Empleado

**Actor Participante:** público

**Resumen:** Mediante esta vista se permitirá el logueo de un usuario empleado al sitio (ingreso de usuario y contraseña). Si el empleado se autentica correctamente, el sistema lo re-direccionará al listado interactivo de Artículos. Luego de logueado el usuario, **todos los accesos a la base de datos, deberán ser a través de su propio usuario de Sql Server (no podrán realizarse acciones con acceso de usuarios públicos)**

Considerar que el menú principal debe contener exactamente los ítems acá propuestos (son exactamente 7 ítems). No se podrán unir acciones sobre conceptos entre sí, ni con listados interactivos propuestos.

### CRUD de Categorías

**Actor Participante:** Empleado

**Resumen:** Permite realizar alta, baja, modificación y consulta individual de Categorías. Se deberá saber un código compuesto de 6 letras/números (identificador único en el sistema) y su nombre. **No se pueden eliminar categorías que tengan artículos asociados.**

*En el listado general se deberá poder filtrar por parte del nombre.*

Por ejemplo: Categoría (“A01MU2”, “Analgésico Muscular”)

### CRUD Artículos

**Actor Participante:** Empleado

**Resumen:** Permite realizar alta, baja, modificación y consulta individual de artículos que están a la venta. Se deberá saber: Código (identificador único en el sistema) compuesto por 10 letras/números, nombre, precio, tipo de presentación, tamaño y fecha de vencimiento. Los tipos de presentación son fijos, a saber: Unidad, Blíster, Sobre o Frasco. Los tamaños son numéricos y deben ser acordes a los tipos de presentación. A saber: Unidad (tamaño: cantidad de unidades), Blíster (tamaño: cantidad de pastillas), Sobre (tamaño: cantidad de gramos), Frasco (tamaño: cantidad de ml). **No se pueden eliminar artículos que tengan ventas asociadas**

*En el listado general se deberá poder filtrar por parte del nombre.*

Por ejemplo: Artículo (“ANGRI10239”, “Anti Gripal Diurno”, 130, Sobre, 45, Categoría(...))

### Mantenimiento de Clientes

**Actor Participante:** Empleado

**Resumen:** Permite realizar alta, modificación y consulta individual de clientes. Se deberá saber: número de CI (identificador único en el sistema), nombre, número de tarjeta con la cual realizará los pagos de las ventas, y teléfono de contacto. **No se contempla la posibilidad de eliminar clientes del sistema.**

*En el listado general se deberá poder filtrar por parte del nombre.*

Por ejemplo: Cliente (27007003, “Juan Gomez”, 1234567890123456, 099858585)



### Alta Venta

**Actor Participante:** Empleado

**Resumen:** Permite realizar la venta de artículos. El usuario deberá seleccionar el/los artículos a vender, ingresar la cantidad que se vende por cada uno, seleccionar el Cliente que paga e ingresar la dirección a la cual se envía la mercadería. Además, se deberá almacenar el número de venta (identificador único y autogenerado por el sistema), la fecha de la venta (automática del sistema) y el monto total de la misma (deberá calcularse automáticamente). Se deberá definir el primer estado de la venta (por defecto el estado es **Armado**), el cual deberá tener fecha y hora de asignado (la misma que la creación de la venta). ***El cliente deberá poderse seleccionar desde una lista desplegable.***

### Seguimiento de Venta

**Actor Participante:** Empleado

**Resumen:** Permite cambiar el estado de una venta. Los estados validos de una venta se describieron en los requerimientos de implementación. El cambio de estado NO podrá ser arbitrario, y deberá seguir el orden propuesto (basado en los tipos de estado que ya se tienen en la base de datos, y el orden es dado por su número). Es decir que una venta en estado *Armado*, solo podrá cambiar a estado *Envío*; si está en estado *Envío*, solo podrá cambiar a estado *Entregado*, etc. Para realizar el cambio de estado, el sistema despliega una lista de las ventas. Dicha lista podrá filtrarse por fecha o estado. El empleado selecciona una venta, y el sistema permite cambiar el estado. Se deberá mantener el histórico de estados de cada venta, teniendo en cuenta que, para cada cambio, se almacena la fecha y hora de realizado (automático en el momento de persistirlo)

### Listado Interactivo de Artículos

**Actor Participante:** Empleado

**Resumen:** Se desplegarán todos los artículos del sistema, ordenados por nombre.

Se podrán realizar filtros por:

- Nombre de artículo.
- Categoría a la que pertenece el artículo

Por cada artículo, se podrán realizar las siguientes acciones

- desplegarse en otra vista todas las ventas asociadas (número – fecha) ordenadas por fecha
- mostrar todos los datos completos de dicho artículo; incluyendo nombre de categoría, tipo de presentación (en letras, no números), y de forma adecuada el tamaño.

***Uso obligatorio de Linq para realizar filtros***

### Listado Interactivo de Clientes

**Actor Participante:** Empleado

**Resumen:** Se desplegarán todos los clientes del sistema, ordenados por nombre.

Se podrán realizar filtros por:

- Nombre de cliente.

Por cada cliente, se podrán realizar las siguientes acciones

- desplegarse en otra vista todas las ventas asociadas (número – fecha) ordenadas por fecha
- listado completo de artículos que compro, pero sin repetidos y ordenados alfabéticamente por nombre (es decir solo interesa saber que artículos compro, no cuantas veces)
- 

***Uso obligatorio de Linq para realizar filtros y ordenes***



## Tutorías y Entregas

A continuación, se detallan las entregas sugeridas:

- Primera Sesión (*semana del 27/01 al 31/01*):
  - Modelo Conceptual
  - DER
  - Script BD completa (tablas – procedimientos – manejo de usuarios – baja lógica)
  - Biblioteca para el modelo de Entity Framework completa
  - Biblioteca Entidades Compartidas completa
  - Biblioteca Persistencia completa
  
- Segunda Sesión (*semana del 17/02 al 21/02*):
  - Biblioteca Lógica completa
  - UI: Sitio Web de ASP.Net completo
  - UI: aplicación web MVC (general y logueo)
  
- Tercera Sesión (*semana del 10/03 al 14/03*):
  - Entrega de todo lo solicitado en el punto **Requerimientos de Entrega**.
  - **Defensa obligatoria del proyecto - ONLINE**. La instalación del sistema deberá respetar la arquitectura propuesta.