

Objetivo:

Hacer un sistema que permita **calcular el monto a cobrar de los empleados** de una empresa **según las horas trabajadas**.

Para ello, **utilizar el proyecto adjunto**, y con la **clase “Empleado”** seguir los siguientes pasos:

1. En la **Consola** del administrador de paquetes, ejecute el comando siguiente:
Install-Package Microsoft.EntityFrameworkCore.Design
2. Localice el **model Empleados**, y **agregar el using** correspondiente para el siguiente espacio de nombres:
System.ComponentModel.DataAnnotations
3. Localice el **model Empleados** y **agregarle el decorado** [Required] **únicamente** para los campos: **Legajo**, **Apellido**, **Nombre** y **HorasTrabajadas**
4. Use la herramienta de **scaffolding** para generar las páginas Create, Read, Update y Delete (**CRUD**) para el modelo de empleado.
5. Luego, vaya al **archivo appSettings.json**, y modifique la entrada correspondiente a la cadena de conexión para que quede con el siguiente nombre la base de datos:
"Server=(localdb)\mssqllocaldb;Database=RecuperatorioContext;Trusted_Connection=True;MultipleActiveResultSets=true"
6. En el menú **Herramientas**, seleccione Administrador de **paquetes NuGet > Consola** del Administrador de paquetes.
 - a. En la Consola del Administrador de paquetes (PMC), escriba el siguiente comando:
Add-Migration InitialCreate
 - b. En la Consola del Administrador de paquetes (PMC), escriba el siguiente comando:
Add-Migration Update-Database
7. En la vista **Shared/_Layout.cshtml**, modifique la entrada del menú principal para **que diga “Empleados App”** y **apunte al controlador del modelo de empleados**:

```
<headers>
<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
  <div class="container">
    <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Empleados App</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
      <ul class="navbar-nav flex-grow-1">
```

8. Ahora, se pide **que el cálculo para el Total A Cobrar se realice desde el controlador mediante la fórmula: Total A Cobrar = Cantidad de horas trabajadas X precio de la hora**, y **no lo ingrese el usuario desde la vista**, para ello, **localizar la vista del controlador de Empleado**, para el Action Method “Create” y busque el elemento html para la propiedad “TotalAcobrar” y **agregarle el atributo style=”display:none”**, además, **en el elemento input**, **agregarle el atributo value=”0”** de esta manera el campo **quedará oculto** para luego ser sobrescrito en el controlador durante el guardado.

```
<div class="form-group" style="display:none">
  <label asp-for="TotalAcobrar" class="control-label"></label>
  <input asp-for="TotalAcobrar" class="form-control" value="0" />
  <span asp-validation-for="TotalAcobrar" class="text-danger"></span>
</div>
```

9. **Modificar el Action Method para el método “Create”** (que recibe el binding del modelo) y **modificarlo para que calcule el campo TotalAcobrar** con la siguiente fórmula:
TotalAcobrar = Cantidad de horas trabajadas * 250

```
public async Task<IActionResult> Create([Bind("
{
    if (ModelState.IsValid)
    {
        empleado.TotalAcobrar /*Completar*/

        _context.Add(empleado);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(empleado);
}
```
10. **Usar la misma lógica de los puntos 8 y 9** (para ocultar y calcular el total a cobrar) tanto **en la vista** como **en el controlador** para la acción **Edit**