

El siguiente procedimiento, describe el paso a paso y contiene la información suficiente para el tener un orden básico de ejecución y administración del proyecto.

El contenido del mismo es genérico y para todos igual. Cada equipo debe trabajando sobre cada uno de ellos, generando nuevos issues, según sea necesario según las necesidades de cada equipo.

**IMPORTANTE!!!** Por favor, cargar todos los issues de una vez, no hacerlo en tramos, esto es para que los id de los issue sean correlativos e iguales entre todos los repositorios. Esta tarea debe ser realizada por un solo alumno por grupo.

## Milestones

Issues > Milestones > New milestone

Title	Due Date	Description
EP1-Modelos	CURSO C- Viernes 7 de Abril CURSO D- Viernes 14 de Abril	Proyecto MVC base, con los modelos y propiedades detectadas en el enunciado y/o detectadas por el grupo.
EP2-Relaciones y Restricciones	Viernes 21 de Abril	Carga o ajustes de propiedades relacionales entre los modelos. Ajustes en las propiedades. Todos los modelos a persistir, deben tener la propiedad Id, Email, UserName. Solo se permite y requiere una herencia en estos modelos y es con fines académicos. (Persona/Empleado, cliente, etc. o según corresponda al proyecto). Carga de restricciones sobre toda propiedad que así lo requiera. Cada restricción debe tener un mensaje de error en castellano y debe hacer sentido a la misma. Nota: Considerar que no se espera que los alumnos hagan Copy/Paste, sino que se detengan a evaluar o re-evaluar los tipos de datos definidos en las propiedades y las restricciones que desean aplicar.
EP3-ABMs	Viernes 5 de Mayo	Agregar y configurar EntityFramework con Database In Memory, y paquete de Identity. Herencia de IdentityUser. Cada entidad a persistir, debe tener generado su ABM básico con Scaffolding y remediar escenarios base que no contempla la generación de código automático. Nota: El fin de este entregable es que tengan un primer contacto con las ABMs(CRUD), visualicen opciones de carga y manejo de datos entre las vistas y los controladores, y puedan ir evaluando necesidades o falencias en estos procesos, que no sean necesarios o válidos en sus WebApps.
EP4-Funcionalidades Básicas	Viernes 1 de Junio	En este entregable se debe Adecuaciones para soportar Identity. Modificar el Database Provider a SQL, agregar migraciones. Adecuar la WebApp para poder realizar las funcionalidades básicas de cada WebApp, adicionalmente a las ABM básicas que puedan quedar y ser suficientes. Ej. Siembre tendré que poder realizar al menos una registración online, Iniciar sesión, cerrar sesión. En un carrito de compras, comprar; en un foro, postear, y así sucesivamente. Para esta entrega es aceptable que las validaciones de negocio no funcionen. Ej, reservar una función cuando no hay butacas disponibles, o comprar un producto que no hay stock, etc.
Presentación de TP	CURSO C – Miércoles 7 de Junio Curso D – Martes 6 de Junio	Muestra de TP en el estado que se encuentra a sus pares. Presentación de 15 minutos, no es necesario PPT ni revisión de código. Solo presentar funcionalmente. Corte de control, solicitado, realizado, pendientes, desafíos encarados, etc.
TP Finalizado	Viernes 16 de Junio	El TP debe estar finalizado al menos en el 80% para considerarse aprobado en la cursada.  Presentación - Semana 16 (del 14 de noviembre al 17) - Fecha a definir  Segunda posibilidad de entrega y ultima 19/11/2022 para su revisión dentro de la cursada. Aquellos que entreguen el proyecto el mismo día de la defensa, automáticamente quedarán con menor prioridad en orden de evaluación. ¿Qué significa esto?, el criterio de llamado de alumnos a la defensa será: 1. Aquellos grupos que entregaron en primera fecha hasta el 12 de noviembre. 2. Aquellos grupos que entregaron hasta fecha 19 de noviembre. 3. Aquellos grupos que entregan el mismo día de la defensa.
Defensa Cursada	21 de Junio	Aquellos grupos que han presentado y aprobado el TP al menos en un 80%, podrán defenderlo. Alumnos que durante la cursada subieron código referente a Modelos, Vistas, Controladores y Persistencia, podrán hacer uso del beneficio de defensa exclusiva de su código. Caso contrario, deberán defender cualquier pregunta del proyecto presentado.
1er Recuperatorio	CURSO C – Miércoles 28 de Junio Curso D – Martes 27 de Junio	Aquellos grupos que entregaran el tp hasta el 23 de Junio tendrán la posibilidad de defensa de la cursada en concepto de recuperatorio.
Final y 2do recuperatorio	Fecha a definir entre la primer y segunda semana de Julio	Aquellos que aprobaron la cursada con el 100% del proyecto, podrán presentarse a defender cualquier pregunta del mismo o cualquier tema visto en la materia.  Aquellos que aprobaron con el 80% del proyecto, deben si o si finalizar el proyecto y presentarlo el sábado anterior a la fecha de final y notificar al docente para evaluarlo.  Aquellos que tengan pendiente de aprobar la cursada, deben si o si presentar el proyecto al 80% y el sábado anterior a la fecha de final y notificar al docente para evaluarlo, y poder defender en fecha de final el recuperatorio. En caso de aprobar, deberán rendir final en el llamado de diciembre.

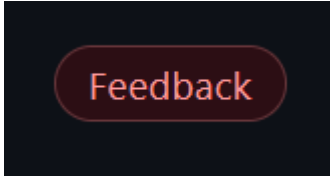
## Labels (Alumno)

Issues > Labels > New label

Label name : Feedback

Color: #b60205

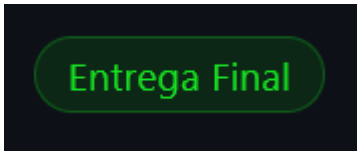
> Create label



Label name : Entrega Final

Color: #0E8A16

> Create label



## Issues (Alumno)

La carga de todos los issues, debe asignarlos al proyecto previamente cargado y el milestone que corresponda.

Importante: Crear todos los issues en el orden especificado, para que los ids. De estos se correlacionen con el Title, y puedan ser accedidos directamente por URL.

Por cada issue nuevo, en la columna de la derecha, asociar a los participantes del grupo, seleccionar su proyecto y el milestone que figure en la tabla.

- Issues > New issue
- Completar todos los datos indicados en la tabla (una fila por issue) > Submit new issue

Title	Comment	Milestone
01.Planificación	<p><b>**Todos los integrantes:**</b></p> <p>- [ ] Definir que integrante es <input type="text" value="123"/> y <input type="text" value="4"/></p> <p>Modelos</p> <p>- [ ] Definición de clases a crear</p> <p>- [ ] Repartirse las clases por integrante</p> <p>Propiedades</p> <p>- [ ] Definir relación y cardinalidad entre clases (idea inicial)</p> <p>- [ ] Definir dependencias</p>	EP1-Modelos
02.Creación de proyecto	<p>Integrante <input type="text" value="1"/> con participación/presencia del equipo:</p> <p>- [ ] Generar proyecto</p> <p>- [ ] Commit con mensaje "Creación de proyecto base"</p> <p>- [ ] Push al repositorio en branch principal</p> <p>Integrantes <input type="text" value="23"/> y <input type="text" value="4"/></p> <p>- [ ] Pull por <input type="text" value="2"/></p> <p>- [ ] Pull por <input type="text" value="3"/></p> <p>- [ ] Pull por <input type="text" value="4"/></p>	EP1-Modelos
03.Generación de Modelos	<p>Integrante <input type="text" value="2"/> con participación/presencia del equipo:</p> <p>- [ ] Generar Clases base (sin propiedades)</p> <p>- [ ] Commit con mensaje "Generación de clases base"</p> <p>- [ ] Push al repositorio en branch principal</p> <p>Integrantes <input type="text" value="13"/> y <input type="text" value="4"/></p> <p>- [ ] Pull por <input type="text" value="1"/></p> <p>- [ ] Pull por <input type="text" value="3"/></p> <p>- [ ] Pull por <input type="text" value="4"/></p>	EP1-Modelos

04.Propiedades	<p>Cada integrante debe trabajar sobre sus propias clases:</p> <ul style="list-style-type: none"><li>- Declarar propiedad Id</li><li>- Declarar propiedades adicionales</li><li>- Declarar tipos de dato para cada propiedad</li><li>- Commit con Mensaje "Propiedades para clases xxx,yyy,zzz"</li><li>- Push</li></ul> <p>_Nota: Se recomienda que en esta instancia, aún no trabajen más de un integrante sobre una misma clase, para no tener problemas de conflictos al momento de hacer un push.</p> <ul style="list-style-type: none"><li>- [ ] Integrante<sup>1</sup></li><li>- [ ] Integrante<sup>2</sup></li><li>- [ ] Integrante<sup>3</sup></li><li>- [ ] Integrante<sup>4</sup></li></ul> <p><a href="https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/auto-implemented-properties">https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/auto-implemented-properties</a></p>	EP1-Modelos
----------------	---	-------------

05. Planificación	<p>Todos los integrantes:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Revisión, Distribuir tareas y remediar según Feedback anterior</li> <li>- Revisan y re-definen</li> <li>- <input type="checkbox"/> Mejoras en las relaciones entre clases</li> <li>- <input type="checkbox"/> Propiedades</li> <li>- <input type="checkbox"/> Clases intermedias</li> <li>- Restricciones</li> <li>- <input type="checkbox"/> Tipos</li> <li>- <input type="checkbox"/> Valores, Rangos</li> <li>- <input type="checkbox"/> Mensajes de error</li> </ul>	EP2-Relaciones y Restricciones
06. Tareas comunes	<p>Integrante <input type="checkbox"/> con participación/presencia del equipo:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Genera clases adicionales (Detectadas o Solicitadas)</li> <li>- <input type="checkbox"/> Clases de soporte (Helpers)</li> <li>- <input type="checkbox"/> Configuraciones comunes</li> <li>- <input type="checkbox"/> Commit "Configuraciones base del EP2"</li> <li>- <input type="checkbox"/> Push</li> </ul> <p>Integrantes <input type="checkbox"/> <input type="checkbox"/> y <input type="checkbox"/></p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Pull por <input type="checkbox"/></li> <li>- <input type="checkbox"/> Pull por <input type="checkbox"/></li> <li>- <input type="checkbox"/> Pull por <input type="checkbox"/></li> </ul>	EP2-Relaciones y Restricciones
07. Ejecución de Mejoras en Relaciones y Restricciones	<p>Todos los integrantes aplican mejoras definidas previamente, cada uno en las clases bajo su responsabilidad para este entregable.</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Push por <input type="checkbox"/></li> <li>- <input type="checkbox"/> Push por <input type="checkbox"/></li> <li>- <input type="checkbox"/> Push por <input type="checkbox"/></li> <li>- <input type="checkbox"/> Push por <input type="checkbox"/></li> </ul>	EP2-Relaciones y Restricciones

08. Planificación	<p><b>**Todos los integrantes:**</b></p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Revisión, Distribuir tareas y remediar según Feedback anterior</li> <li>- <input type="checkbox"/> Imaginar interacción del usuario con el sistema - modalidad de uso, asistente (Wizard, paso a paso), AIO (All-In-One). Básicamente si se quiere que el usuario cargue todo en un único formulario de una vez cuando sea posible, o el sistema lo va guiando por pantallas, carga datos, siguiente, así sucesivamente. No obstante, considerar las funcionalidades que se piden en el enunciado. Si indica modo wizard, debe respetarse.</li> <li>- <input type="checkbox"/> Definición inicial de interfaz, componentes y texto participante</li> <li>- <input type="checkbox"/> Repartirse tareas</li> </ul>	EP3-ABMs
09. Resolución de issues	<p>Todos los integrantes, ejecutarán las tareas de remediación planificadas en base al feedback.</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Integrante <input type="text" value="1"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="2"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="3"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="4"/></li> </ul>	EP3-ABMs
10. Tareas comunes	<p>Integrante <input type="text" value="4"/> con participación/presencia del equipo:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Dependencias (Cargar las dependencias de paquetes solicitadas)</li> <li>- <input type="checkbox"/> Tareas y configuraciones comunes Carpetas Data / EntityFramework / Startup / etc.</li> <li>- <input type="checkbox"/> Commit "Configuraciones base EP3"</li> <li>- <input type="checkbox"/> Push</li> </ul>	EP3-ABMs
11. Creación de ABMs	<p>Todos los integrantes generan las mejoras definidas y creación de los ABMs de las clases bajo su responsabilidad.</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Pull, Push y Merge a develop por <input type="text" value="1"/></li> <li>- <input type="checkbox"/> Pull, Push y Merge a develop por <input type="text" value="2"/></li> <li>- <input type="checkbox"/> Pull, Push y Merge a develop por <input type="text" value="3"/></li> <li>- <input type="checkbox"/> Pull, Push y Merge a develop por <input type="text" value="4"/></li> </ul>	EP3-ABMs

12. Planificación	<p>Todos los integrantes:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Revisión, Distribuir tareas y remediar según Feedback anterior</li> <li>- <input type="checkbox"/> Revisión y definición de prioridades en las funcionalidades</li> <li>- <input type="checkbox"/> Repartirse tareas</li> </ul>	EP4-Funcionalidades básicas
13. Resolución de issues	<p>Todos los integrantes, ejecutarán las tareas de remediación planificadas en base al feedback.</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Integrante <input type="text" value="1"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="2"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="3"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="4"/></li> </ul>	EP4-Funcionalidades básicas
14. Tareas comunes	<p>Integrante <input type="text" value="1"/> con participación/presencia del equipo:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Dependencias (Instalar paquetes necesarios)</li> <li>- <input type="checkbox"/> Tareas y configuraciones comunes - Identity</li> <li>- <input type="checkbox"/> Commit "Configuraciones base EP4"</li> <li>- <input type="checkbox"/> Push</li> </ul>	EP4-Funcionalidades básicas
15. Identity management	<p>Agregar las siguientes funcionalidades:</p> <p>Usuarios:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Registración (auto-registración para entidad que corresponda)</li> <li>- <input type="checkbox"/> Login</li> <li>- <input type="checkbox"/> Logout</li> </ul> <p>Roles - Modificaciones e implicancias en:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> En registración (asignación de roles automatico, dependiendo la necesidad)</li> <li>- <input type="checkbox"/> En Creación (asignación de roles automatico, dependiendo la necesidad)</li> <li>- <input type="checkbox"/> En Acceso (Restringir acceso basado en roles)</li> <li>- <input type="checkbox"/> En Layout (Adecuar el layou, para mostrar diferentes funcionalidades de acceso, basado en rol)</li> </ul>	EP4-Funcionalidades básicas
16. Funcionalidades básicas	<p>Completar funcionalidades básicas definidas en la planificación, según orden de prioridad. Recordar que falta de validaciones o errores en consistencias de datos, pueden ser permitidos en esta instancia. Ej. Poder alquilar una cochera, cuando todas las cocheras ya están tomadas.</p>	EP4-Funcionalidades básicas

17. Planificación	<p>Todos los integrantes:</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Revisión, Distribuir tareas y remediar según Feedback anterior</li> <li>- <input type="checkbox"/> Revisión y definición de prioridades en las funcionalidades</li> <li>- <input type="checkbox"/> Repartirse tareas</li> <li>- <input type="checkbox"/> Planificación inicial de Presentación</li> </ul>	TP Finalizado
18. Resolución de issues	<p>Todos los integrantes, ejecutarán las tareas de remediación planificadas en base al feedback.</p> <ul style="list-style-type: none"> <li>- <input type="checkbox"/> Integrante <input type="text" value="1"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="2"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="3"/></li> <li>- <input type="checkbox"/> Integrante <input type="text" value="4"/></li> </ul>	TP Finalizado
19. Ajustes de Funcionalidades básicas	Realizar ajustes a las funcionalidades básicas definidas en la planificación según orden de prioridad. Recordar que para aprobar el proyecto durante la cursada se requiere el ~80% de las funcionalidades completadas.	TP Finalizado
20. Funcionalidades completas	Completar la totalidad de funciones solicitadas.	TP Finalizado
21. Checklist de evaluación	Este checklist será cargado por el docente según cada proyecto	TP Finalizado