

# Proyecto Parte 1: Aplicación web tradicional

## Objetivos

### Objetivo general

Construir una **aplicación web tradicional con un propósito específico** utilizando el framework Koa para Node.js. Esta aplicación web estará definida en base a una idea de proyecto asignada al equipo de trabajo, y tendrá funcionalidades simples que permitan una interacción fluida con el usuario, de manera tal que los estudiantes obtengan como resultado una aplicación web que perfectamente podría ser utilizada por usuarios reales.

### Objetivos específicos

- Utilizar la arquitectura MVC para la construcción de una aplicación web tradicional, con ayuda de un template pre-definido construido específicamente para el curso
- Implementar manejo de usuarios (registro y sesiones)
- Diseñar la interfaz de usuario de la aplicación en base a una idea de proyecto asignada
- Implementar un frontend básico utilizando los lenguajes HTML y CSS
- Incorporar aspectos de calidad en relación a código y metodologías de desarrollo

## Contexto

Actualmente nos encontramos en una época en que nuestras interacciones se llevan a cabo mayoritariamente de forma digital, al punto en que la Web se ha transformado en una plataforma para organizarse y planificarse de forma personal y/o en comunidad con un objetivo específico.

En este proyecto los estudiantes tendrán la oportunidad de crear una aplicación web en base a una idea de proyecto particular, que de forma general consiste en una plataforma destinada a satisfacer una necesidad personal y/o grupal. Las ideas de proyecto incluyen una base que los estudiantes deben respetar, sin perjuicio de permitirles profundizar en el desarrollo de la idea.

## Ideas de proyecto

Este semestre existen 5 ideas disponibles, que fueron seleccionadas a partir de ideas propuestas por ustedes mismos. A cada equipo de trabajo se le asignará una de estas ideas para que la desarrollen durante el semestre. Si bien tendrán la posibilidad de comunicar sus preferencias, que por supuesto serán consideradas, las ideas de proyecto serán distribuidas de forma equitativa entre todos los grupos de trabajo de ambas secciones.

Las 5 ideas de proyecto seleccionadas son:

1. [BeachWeekWeb](#)
2. [LaPerApp](#)
3. [MemeMatch](#)
4. [UrbanGarden](#)
5. [WeWOD](#)

El detalle de cada idea de proyecto lo podrán encontrar en el documento independiente enlazado en cada uno de los puntos anteriores.

## Criterios de diseño

Los equipos de trabajo deben proponer y generar una solución de acuerdo a la idea de proyecto que les sea asignada. Esta solución debe cumplir con los criterios descritos a continuación:

1. La solución propuesta debe ser adecuada en interés y complejidad para el contexto de una aplicación web tradicional (i.e. seguir un diseño de estilo MVC accesible desde un browser)
2. La solución propuesta debe incluir un flujo de navegación que el usuario pueda seguir (o al menos debe poder extraerse a partir de lo propuesto), que incluya diferentes vistas (las que se traducirán en rutas en la aplicación)
3. Buenas prácticas que se verán como contenido del curso en cátedras y ayudantías

## Entregas

Esta primera parte del proyecto se compone de **3 entregas**: una sin nota (0) y dos con nota (1 y 2). Las entregas estarán separadas una de otra por un intervalo cercano a 2 semanas, finalizando justo antes de la semana de receso de este semestre.

En este documento encontrarán lineamientos generales de la funcionalidad que deben implementar en cada una de las entregas, pero serán deliberadamente ambiguos de manera que ustedes usen su creatividad para crear una aplicación realmente única. Salvo algunos casos, ustedes tendrán total libertad en cuanto a la interfaz y la implementación de su trabajo.

## Fechas de entregas

- **Entrega 0:** 14 de septiembre (evaluada sin nota)
- **Entrega 1:** 30 de septiembre
- **Entrega 2:** 14 de octubre

## Forma de entrega

Todo el código será entregado únicamente a través de un **repositorio Git alojado en Github** y asociado a su grupo. El equipo docente les proveerá a los estudiantes repositorios gratuitos y privados para cada equipo.

**Los ayudantes revisarán el branch main únicamente.** Deben usar otras ramas durante el desarrollo y realizar merge a main sólo de lo que están seguros de querer entregar. Para la Entrega 2 tendrán que seguir un flujo definido para hacer merge a main, mediante el uso de Pull Requests. El detalle se encuentra explicado en la sección correspondiente a esa entrega.

Los estudiantes pueden incluir un archivo README en el repositorio del proyecto con cualquier consideración, supuesto, decisiones de diseño u otra información relevante que pueda ser importante para el ayudante que los evaluará. Este README puede ser actualizado con cada entrega.

Para alojar sus aplicaciones en la Web, la plataforma soportada y recomendada será **Heroku**. Si los estudiantes desean utilizar otro proveedor pueden hacerlo, pero será bajo su propia responsabilidad.

El resultado final esperado es que tengan que realizar un deploy real de su aplicación en modo producción y que esta quede disponible para cualquier usuario de la Web, es decir, que sea pública. **No cumplir con este requisito es equivalente a no haber entregado su trabajo, lo cual se traducirá en un 1.0 como calificación.** La única excepción será la Entrega 0, donde sólo será necesario dejar disponibles los archivos en el repositorio correspondiente.

El avance logrado en cada entrega se evaluará en base a **lo subido al branch main del repositorio Github en la fecha y hora límite de la entrega**, y a **lo mostrado en Heroku** en el momento de la reunión con el ayudante. Se verificará que el último commit en Heroku (o la plataforma que hayan seleccionado) corresponda al último commit en Github antes de la fecha y hora límite.

## Entrega 0

### Objetivo: Diseño solución y sitio web estático (HTML + CSS)

**Fecha de entrega:** 14 de septiembre a las 22:00 horas

En esta primera entrega los estudiantes deben concentrarse en el diseño de la solución y en el frontend de la aplicación. Se les recomienda imaginar la aplicación finalizada, y pensar cómo les gustaría que funcionase, incluyendo distintas opciones de presentación y una forma adecuada de incorporar la funcionalidad de su proyecto.

En este momento no tendrán que implementar funcionalidad aún, sino que deben construir lo siguiente:

1. Un **sitio web estático, simulando las páginas principales de la aplicación final**. “Simular” se refiere en este caso a implementar páginas web ficticias y fijas (siempre se verá lo mismo cada vez que se ingrese al sitio web).
2. Un **modelo de datos preliminar** en que aparezcan las principales entidades con los atributos más relevantes y la forma en que están asociados (1 a 1, 1 a N o N a N).

Para el primer punto deben crear un layout (estructura) común para el sitio web y que sea compartido por todas las páginas. En general un layout incluye: header, navegación, contenido y footer. El contenido, por supuesto, es lo que varía de una página a otra, pero el resto se mantiene igual. Se puede hacer el supuesto de que un usuario ya inició sesión (y sería conveniente “nombrar” a este usuario ficticio).

En concreto, el requisito es **construir las siguientes 3 páginas web**, referenciadas (“linkeadas”) entre sí:

- **Colección recurso principal:** la lista genérica del recurso principal de la plataforma. Se debe simular un número razonable de elementos (más de 10). El header del sitio web debe incluir algún link al inicio, que al ser presionado, lleve a esta página con la colección.
- **Detalle recurso principal:** se requiere una sola página web. Como cada recurso debería tener su propia página, sería poco eficiente en esta etapa escribir una página web por cada uno, por lo que en este caso, todo link a un recurso puede referenciar esta única página web de detalle
- **Perfil de usuario:** incluye la información pública del usuario. La forma de acceder a esta página de perfil de usuario es haciendo click en un link que se debe encontrar en el header del sitio web (puede ser el nombre del usuario o algún ícono)

Para el segundo punto, los estudiantes deben **representar su modelo de datos con un diagrama ER (Entidad Relación)**. Esto les permitirá tener algo intermedio que luego podrán mapear tanto a tablas de bases de datos como a un modelo orientado a objetos. Deben apuntar a capturar lo más que puedan, pero no es necesario que sea absolutamente exhaustivo (muy probablemente cambiará durante el desarrollo del proyecto). Se deben preocupar de incluir los atributos que obviamente van a necesitar, por lo que se les aconseja privilegiar la simplicidad cuando sea posible.

Hay muchas herramientas para generar este tipo de diagramas y pueden utilizar la que más les guste. Algunos ejemplos:

- <https://erdplus.com>
- <https://www.diagrams.net/>

Si bien esta entrega no tiene una nota asociada, el completar lo solicitado les ayudará bastante a los estudiantes a no “arrastrar” trabajo hacia las entregas que sí tienen nota asociada. **Todo el trabajo que realicen en esta entrega puede ser reutilizado en la aplicación final.**

Para esta entrega, los estudiantes deben utilizar el repositorio indicado por su ayudante. **Deben incluir todos los archivos de su entrega, incluyendo un PDF con el modelo de datos, dentro de una carpeta llamada “E0”.**

Cualquier duda general que tengan los estudiantes sobre el trabajo a realizar puede ser consultada en el foro del curso. Cualquier duda particular sobre su proyecto, pueden consultarla directamente a su ayudante. Se hace hincapié en que **toda duda la expresen temprana y oportunamente**, pues es muy difícil tratar de manera adecuada issues o preguntas con plazos muy cercanos a la fecha de entrega.

Finalmente, es recomendable que trabajen en grupo y no en forma individual repartiéndose el trabajo, pues de esta manera aprenderán parte importante de lo que ocurre en el frontend. Sin embargo, se tomará en consideración que **todos los integrantes del equipo de trabajo tengan commits en el repositorio** (en caso contrario, existirá penalización que podría llegar a la nota mínima para algún integrante).

## Entrega 1

### Objetivo: CRUD recurso principal

**Fecha de entrega:** 30 de septiembre a las 22:00 horas

En esta segunda entrega los estudiantes deberán producir **la primera versión funcional de la aplicación, utilizando el template del curso**. En términos generales lo que se les pide es que tengan un CRUD (Create, Read, Update and Delete) del recurso principal y de los usuarios, y que opcionalmente utilicen una asociación entre ambos.

En esta entrega **no es necesario** que la aplicación cuente con un manejo de sesión de usuarios. Las acciones que necesiten de un "usuario actualmente logueado", en caso de haber implementado la asociación con el recurso principal, se pueden implementar mientras tanto con una selección del mismo en el formulario de creación.

En este punto no es importante que las páginas se vean del todo acabadas, aunque sería muy deseable que los estudiantes al menos pudieran respetar el layout que implementaron en la Entrega 0.

Para esta entrega, los estudiantes deben utilizar el mismo repositorio indicado por su ayudante para la Entrega 0.

Cualquier duda general que tengan los estudiantes sobre el trabajo a realizar puede ser consultada en el foro del curso. Cualquier duda particular sobre su proyecto, pueden consultarla directamente a su ayudante. Se hace hincapié en que **toda duda la expresen temprana y oportunamente**, pues es muy difícil tratar de manera adecuada issues o preguntas con plazos muy cercanos a la fecha de entrega.

Finalmente, es recomendable que trabajen en grupo y no en forma individual repartiéndose el trabajo, pues de esta manera aprenderán todo el proceso inicial de crear una aplicación web utilizando el template del curso. Sin embargo, se tomará en consideración que **todos los integrantes del equipo de trabajo tengan commits en el repositorio** (en caso contrario, existirá penalización que podría llegar a la nota mínima para algún integrante).

## Entrega 2

### Objetivo: Aplicación final + aspectos de calidad

**Fecha de entrega:** 14 de octubre a las 22:00 horas

En esta última entrega de la primera parte del proyecto deben **finalizar su aplicación web** en cuanto a funcionalidades, incorporando manejo de sesión de usuarios y de todas las acciones que estos puedan realizar en la plataforma. Además, se evaluarán aspectos de calidad de software específicos que serán detallados a continuación.

Los requerimientos generales son los siguientes:

- Registro de usuarios y manejo de sesión (sign in/out), modificando funcionalidad existente considerando el "current user".
- Todas las funcionalidades que hayan negociado con su ayudante
- Validaciones de servidor asociadas a datos y acceso de usuarios, donde sea pertinente.
- Vistas y estilos finales con CSS (SCSS). Tener en consideración que **no se deben utilizar estilos inline**.
- Cualquier funcionalidad extra que los estudiantes decidan implementar (siempre que no altere la funcionalidad requerida para su idea de proyecto y mantenga la coherencia). No olviden usar la guía de sus ayudantes.

Los requerimientos de calidad de software son los siguientes:

- **Metodología de Code Reviews utilizando Pull Requests.** Los repositorios de los estudiantes estarán configurados de tal forma que sólo puedan hacer merge a main a través de la interfaz de *Pull Requests* (no será posible realizar merge directos por consola). Además se requerirá la revisión de código y aprobación de al menos un estudiante del grupo para hacer merge a main
- **Análisis estático de código a través de ESLint.** Los repositorios de los estudiantes estarán configurados para ejecutar ESLint cada vez que se haga un commit. Si el *build* no es exitoso, cualquier Pull Request que incluya ese commit no podrá ser *merged* en main

Para esta entrega, los estudiantes deben utilizar el mismo repositorio de la aplicación web que han utilizado para las entregas anteriores.

Deben considerar, además, que **esta entrega equivale al primer lanzamiento de su aplicación web**, por lo que deben asegurarse de verificar detalles, como por ejemplo, incorporar datos razonables y realistas.

Cualquier duda general que tengan los estudiantes sobre el trabajo a realizar puede ser consultada en el foro del curso. Cualquier duda particular sobre su proyecto, pueden consultarla directamente a su ayudante. Se hace hincapié en que **toda duda la expresen temprana y oportunamente**, pues es muy difícil tratar de manera adecuada issues o preguntas con plazos muy cercanos a la fecha de entrega.

Finalmente, es recomendable que trabajen en grupo y no en forma individual repartiéndose el trabajo, pues de esta manera aprenderán sobre el proceso de finalización de una

aplicación web y su lanzamiento. Sin embargo, se tomará en consideración que **todos los integrantes del equipo de trabajo tengan commits en el repositorio** (en caso contrario, existirá penalización que podría llegar a la nota mínima para algún integrante).

## Evaluación

Tal como está indicado en el programa del curso, cada entrega de proyecto será evaluada con una nota de 1.0 a 7.0 siguiendo una escala discreta que se detalla a continuación:

Calificación	Nota asociada	Observaciones
A	7.0	Trabajo sobresaliente. La entrega excede con creces lo solicitado.
A-	6.0	Muy buen trabajo. La entrega cumple con más de lo solicitado.
B	5.5	Buen trabajo. La entrega cumple con lo mínimo solicitado.
C	4.0	Regular. La entrega cumple parcialmente con lo mínimo solicitado.
D	2.5	Deficiente. La entrega está muy por debajo del mínimo solicitado.

**En el caso de que un grupo no entregue nada, o esté fuera de plazo (habiendo utilizado la totalidad de sus cupones de atraso), la nota será de un 1.0.**

Además, posterior a la calificación de cada entrega grupal, los estudiantes integrantes del grupo deberán calificarse entre sí en una **evaluación de pares**. Pueden revisar los detalles de cómo se llevará a cabo esta evaluación en el programa del curso.

Para evaluar cada entrega, todos los miembros del equipo tendrán una **reunión (virtual o presencial) de corrección** con el ayudante que se les haya asignado. La no asistencia (salvo justificación de peso) de alguno de los integrantes implicará un **descuento de 0.2** para ese estudiante **en el ponderador por evaluación de pares** en esa entrega (no 2 décimas).

Finalmente, las notas de las entregas parciales del proyecto (luego de evaluación de pares) serán **promediadas** y eso generará la calificación final del trabajo práctico, denominada Nota Práctica (NP). Consideren que las entregas 3 y 4 corresponden a la Parte 2 del Proyecto, y el detalle será publicado luego de finalizar esta primera parte.



## Consideraciones y restricciones

- La aplicación web debe desarrollarse en Node.js y koa **utilizando el template del curso**.
- El templating engine soportado será EJS (con HTML), pero si lo desean podrán usar otros lenguajes de templates bajo su responsabilidad.
- Podrán usar SASS/SCSS, less, stylus u otros preprocesadores de CSS.
- NO está permitido el uso de lenguajes que compilen a JavaScript, como CoffeeScript o TypeScript.
- Tampoco está permitido usar librerías de componentes HTML/CSS/JS como Bootstrap, Bulma, Tailwind CSS, etc, salvo una que otra excepción (aquí deberán consultar en el foro). Pero si quieren, ¡vean su código fuente y úsenlos de inspiración!
- **Cualquier *package* o librería adicional que quieran usar debe ser aprobado por el equipo docente. Pregunten a través del foro del proyecto si tienen intención de utilizar alguna.**
- Es recomendable escribir código en inglés. Sacarán el máximo provecho de las convenciones que puedan haber y el código quedará en armonía con el lenguaje de programación que usen (que también estará en inglés).
- La interfaz de la aplicación web puede estar en inglés o en español.