# PROMOCIÓN

# LABORATORIO DE PROGRAMACIÓN Y LENGUAJES – 2018

Alumnos: Brandon Urigo y Ezequiel Silvestre

# ACTIVIDADES

Para el desarrollo de este examen de promoción lo primero que hicimos fue dividir las tareas. Urigo se encargó del desarrollo BackEnd (PHP, Javascript, SQL, Ajax) y Silvestre se encargó del desarrollo FrontEnd (HTML, Javascript, CSS).

Debido a la naturaleza del examen todo el desarrollo se llevó a cabo usando el sistema de colaboración Git, mediante la plataforma Github y los clientes SourceTree (Urigo) y Github Desktop (Silvestre). De esta manera fue mucho más fácil coordinar las tareas

En el siguiente enlace podrán encontrar el repositorio de Github del examen:

<https://github.com/zeqelandia/Promocion-LPL>

En este otro enlace se pueden ver todas las subidas de archivos realizadas a lo largo del desarrollo y el detalle de cada una de ellas:

<https://github.com/zeqelandia/Promocion-LPL/commits/master>

# ACLARACIONES

Para el desarrollo no utilizamos ningún tipo de librerías exteriores a excepción de Animate.css que se encarga de facilitar el manejo de animaciones visuales. Otro recurso utilizado son fuentes de google fonts embebidas.

# PLANEAMIENTO

Antes de comenzar con el código, se analizaron algunos parámetros. Entre estos se encuentran: la consigna, el diseño visual, la lógica y el diseño de la base de datos.

Dentro de la consigna surgieron 5 ambigüedades:

* Número de asiento: se aclara la distribución de los asientos pero no se especifica nada acerca del número de asiento del pasajero, por lo que este número optamos por no designarlo. Sin embargo, si se respeta la distribución y cantidad de asientos según el tipo de tarifa elegida.
* Recorrido del viaje: No se indica si un viaje puede ser de ida o ida y vuelta, por lo que optamos por solo hacer viajes de ida.
* Editar perfil: Se pide que el usuario pueda modificar datos de su perfil. Sin embargo, en nuestro diseño de la base de datos el DNI del usuario sirve de clave primaria, por lo que no es posible para un pasajero modificar su DNI una vez creado el usuario. Teniendo en cuenta esto el sistema informa al usuario en caso de que el dni ingresado ya está en uso al momento de crear la cuenta.
* Selección de boletos: Según qué ciudad de origen se elija, se puede elegir la ciudad de destino. En base a estas dos se elige el horario y la tarifa. Sin embargo, la duda que surgió fue: es necesario implementar las ciudades, las combinaciones de estas y las tarifas acorde a estas, en la base de datos? Esta duda surge debido a que no se especifica en la consigna si serán agregadas, o podrían ser agregadas, más ciudades, ni combinaciones entre estas, ni cambios en las tarifas. Por lo que en vez de manejar esto utilizando la base de datos, optamos por utilizar javascript. Sin embargo, sí se utilizó la base de datos para almacenar los tipos de tarifas, esto debido al diseño de la misma.
* Cantidad de boletos por compra: No se especifica si se puede comprar más de un boleto en la misma compra, por lo que optamos por que solo se pueda comprar de a un boleto.

Para el diseño visual se analizaron varias páginas de viajes y turismo. Si bien se tomaron ideas de otras páginas, no se utilizó ningún tipo de plantilla de terceros al momento del desarrollo. Se trató de lograr una página fácil de utilizar pero atractiva visualmente y acorde a las últimas tendencias de diseño.

Para la lógica hubo charlas previas, para saber que parte de esta pertenecía al frontend y cual al backend y cómo interactúan estas dos. Por ejemplo para la compra de un boleto:

* Frontend: La selección de ciudad de origen, ciudad de destino, horario y tarifa, y la consistencia de estas selecciones se lleva a cabo mediante javascript.
* Backend: Una vez válidos los datos de la compra esta se carga en la base de datos y se actualiza la cantidad de asientos en el colectivo.

El diseño de la base de datos es sencillo, existen 4 tablas: usuarios, viajes, tipo\_tarifa y compras.

* Tipo\_tarifa: almacena los 4 tipos de tarifas existentes, esto debido a que es necesario conocer esta información para cada compra.
* Usuarios: almacena la información relativa a cada usuario. Además almacena la cantidad de kilómetros sumados si el usuario es pasajero frecuente.
* Viajes: almacena información relativa a cada viaje. Cada uno tiene un id único. Esta tabla se implementó para poder llevar un control de asientos por cada viaje.
* Compras: cada compra tiene un id único. Una compra asocia un usuario a un viaje, el tipo de tarifa elegida y el valor abonado. Esta tabla se implementó para poder llevar un historial de compras para cada cliente.

# DESARROLLO

9 de julio:

* Silvestre: se crearon los directorios principales y se añadió la primer versión de la base de datos.

11 de julio:

* Urigo: actualización de la base de datos y se agregó el Ajax al momento de crear una cuenta.
* Silvestre: Se incorporó un logo y nombre a la página para darle mas personalidad. Ademas se definieron las plantillas visuales para botones e inputs.

12 de julio:

* Silvestre: primera versión de index.html. Se agregó una plantilla para los contenedores visuales (div).
* Silvestre: Se agregó la pantalla de login y crear cuenta. Se mejoraron las plantillas y se llevaron a cabo las primeras pruebas con el fondo de la pagina.

13 de julio:

* Silvestre: Se corrigieron errores y se completó el desarrollo de las pantallas de login y crear cuenta.

14 de julio:

* Urigo: Se agregaron cambios a index.html e index.js. Además se creó la clase base de datos y el archivo login.php.

16 de julio:

* Silvestre: Se completó el desarrollo visual de la reserva de boletos y se incorporó una animación para el fondo de la página.
* Urigo: Avances en el login y crear usuario.

17 de julio:

* Silvestre: Se actualizó la base de datos y se implementó la versión completa.

18 de julio:

* Silvestre: se agregó user.html que contiene la ventana que ve el usuario una vez logeado. Además se dio navegación a los apartados de Comprar Boletos, Editar Perfil, Ver Historial de Compras y Cerrar Sesión.
* Urigo: avances en el login y crear usuario.

19 de julio:

* Urigo: Login y Crear usuario totalmente terminados. Además se incorporó la sesión a través de las diferentes páginas y esta también se puede cerrar para ingresar con otro usuario. Modificaciones menores a la base de datos.

20 de julio:

* Silvestre: Se terminó el desarrollo de la ventana historial de compras. Además se añadieron títulos a cada ventana y se añadió un footer en cada una. Se añadieron ventanas de confirmación para algunas operaciones y se incorporó un comprobante al momento de realizar una compra.
* Urigo: Se completó el desarrollo de Editar Usuario y sus respectivas consistencias.

21 de julio:

* Silvestre: Correcciones menores y verificación de login al momento de comprar tickets.
* Urigo: Compra de boletos totalmente terminada.

22 de julio:

* Urigo: Se incorporó la compra con kilómetros (o puntos) para pasajeros frecuentes.

23 de julio:

* Silvestre: Se agregaron los carteles de confirmación y cambios de estado faltantes y se dio navegación a ticket.php. Además se corrigió un bug en el historial de compras que deformaba la página.
* Urigo: Corrección de errores.