

Planificación preliminar y presupuesto (proyecto para Clasing ELTS)

El **proyecto** a desarrollar para Clasing ELTS consistirá en una **aplicación** (web + móvil) de gestión de la actividad relacionada con las labores diarias y la **operativa** de dicha empresa.

En una primera versión se contemplarán todos los **aspectos esenciales** de la actividad, lo que supone:

- **Características de Atención al cliente**

- Panel de gestión total (por definir).
 - ➔ Clases diarias, pagos, clases on live, incidencias, reportings, etc.
- El sistema asignará tareas dependientes de ciertos eventos (cuando ocurren 2 valoraciones negativas, cuando un profesor recibe varios puntos satisfactorios, cada X semanas e-mail o llamada para hacer acción de fidelización, etc.).
- Evaluación de todos los pasos y posibilidad de prepararlos de manera manual (por si fuera necesario).
- Acciones predefinidas (mandar e-mail, preparar una oferta, etc).
- Algoritmos inteligentes:
 - ➔ Profesores (alta/baja, asignación de créditos por clases impartidas).
 - ➔ Clases (reserva, cancelación y reprogramación, asignación de clases a profesores, reasignación de clases a profesores en caso de incidencia con el profesor titular en el momento de ofrecer la clase).
 - ➔ Pruebas.
- El sistema debe generar las tareas a los de CSR y repartirlas en función de si están online.
 - ➔ Fidelización, clase fallida, comprobar estado, cualquier acción o conjunto de acciones de carácter temporal que se defina más adelante.
- Contar con un “Encolador de tareas”.
- Avisos a asesores.

- **Características de Profesores**

- Preparación de la clase (selección del temario, llamada y posterior inclusión del feedback del docente, mediante formulario).
- Gestión de créditos, incluyendo salarios y preparación de recibos.
- Contemplar Supervisores (+ gestión de profesores a su cargo).
- Contemplar Trainers (+ gestión de sus trainers), incluyendo un booking para hacer un training con su posterior feedback.
- Panel de vídeo para trainings.
- Mensaje in-app entre todos los perfiles.
- Panel propio del perfil.
 - ➔ Pedir día libre, día de enfermedad (con sus procesos) o descargar ficheros tipo (confirmación de que trabajan, etc.).

- **Características de Alumnos**

- Proceso de captación.
 - ➔ Registro, gestión de prueba de nivel, venta mediante app, gestión de horario final y pago (obtención de créditos).
- Monedas y ventas dependientes del país.
 - ➔ Panel para gestionar dichas ofertas.
 - ➔ Panel para aplicar promociones y ofertas segmentadas.

- ➔ Recibos o facturas.
- Modo empresa.
 - ➔ Control desde recursos humanos de los bonos, avances, pagos, etc.
- Debe ser Multi-idioma.
- Contacto in-app con atención al cliente.
- Gamificación (mostrar evoluciones del nivel, clases, puntuaciones, exámenes, etc.).
- Notificaciones relevantes mediante dashboard.
- Mensajes in app dirigidos (pop ups con ofertas, info relevante, etc.).
- Auto-gestión de clases y funciones propias del perfil.
- Posibilidad de lanzar “huecos libres” en oferta.
- Cada alumno debe tener un reporting en el que, escriba quien escriba, quede guardado como un timeline (un profesor no debe ver apuntes de atención al cliente si no se desea).
- Estrategia de referidos, donde cada usuario puede venir de una fuente distinta (cuando es un referido o comercial, debe quedar reflejado para pagar las comisiones pertinentes).
- Valoración de las clases por parte de alumnos (feedback).
- **Características del “Modo dios”**
 - Se trata de una versión aumentada de atención al cliente, con más acceso.
- **Características de Recursos Humanos**
 - Crear cuentas de profesor.
 - Mismo acceso que un Supervisor o Trainer.
- **Características de Marketing**
 - Similar a atención al cliente, pero basado en planes, fechas de inicio y fin, información respecto de compras y gasto de créditos, etc.
 - Debe poder marcar al usuario como lead, user, user paid, etc.
 - Almacenar con el sistema de referidos de donde viene cada usuario (si viene de un blog, paid search, etc.).
- **Aspectos generales**
 - Cross domain.
 - Estudiar la posibilidad de usar API de google docs y skype en las lecciones.

Todos los contenidos de la aplicación se producirán respetando los principios de **accesibilidad** definidos en la directiva europea **EN 301 549**, de obligado cumplimiento en un futuro próximo para la creación de productos y servicios TIC en el sector público (o de empresas que reciban fondos públicos de algún modo).

Además, se incluyen **6 meses** de **mantenimiento correctivo** desde la puesta en producción de la aplicación, en los que se deberán aplicar aquellos cambios que sean necesarios para garantizar el **correcto funcionamiento** de todas las características mencionadas anteriormente (sin coste adicional). Sin embargo, este mantenimiento **NO INCLUYE** la incorporación de nuevas funcionalidades o la re-definición de las ya existentes.

A continuación se ofrece un pequeño detalle de las tecnologías a utilizar, una planificación orientativa donde se fijan los hitos principales y una sección final de presupuesto.

Tecnologías del desarrollo

Para el desarrollo descrito se utilizarán, siempre que sea posible, **tecnologías de código abierto** que no requieran un pago por su licencia para uso comercial, lo que decrementa los posibles costes derivados de su utilización y garantiza la presencia de una amplia comunidad para la resolución de problemas (mayor **mantenibilidad**).

Además, todas las tecnologías han sido seleccionadas pensando en la **fluidez** y **flexibilidad**, para que la inclusión de **nuevas funcionalidades** en el futuro suponga el menor impacto posible sobre el sistema inicial.

Grosso modo, las **tecnologías** involucradas en la aplicación son:

1. Base de datos: MongoDB 4.
2. WEB:
 - *Front End*: HTML5, CSS3 + framework Bootstrap 4, javascript + framework Angular 6.
 - *Back End*: PHP 7 + framework Laravel o Node JS.
3. Móvil: framework Ionic (para generar versiones funcionales en Android e iOS).
4. Servidor: para la realización de pruebas durante el desarrollo se utilizará un servidor de preproducción externo (ajeno a Clasing ELTS), evitando así reducir la productividad ofrecida por el servidor propio de la empresa.

En todo momento se intentará utilizar la **versión más reciente** de cada tecnología, por motivos de rendimiento y seguridad.