Olimpiada de Programación – 2025

E.E.T.P. N°480 Manuel Belgrano

Santa Fe de la Vera Cruz - Santa Fe

Profesor Responsable: Apellido y nombre, Título de grado o profesorado, Cargo docente, Correo electrónico, Teléfono celular (con prefijo jurisdicción)

Fantini Fani

Profesora de Enseñanza Superior en Programación

Docente de cátedra

[fantinifani@gmail.com](mailto:fantinifani@gmail.com)

342 4383914

Estudiantes:

* Acosta, Lionel - Informática - Ciclo Superior - 6to año
* Favrat, Matias - Informática - Ciclo Superior - 6to año
* Kettler, Thomas - Informática - Ciclo Superior - 6to año

Etapa 1: Análisis y Planificación

* Relevamiento de necesidades: Entrevista simulada al cliente para obtener los requerimientos.

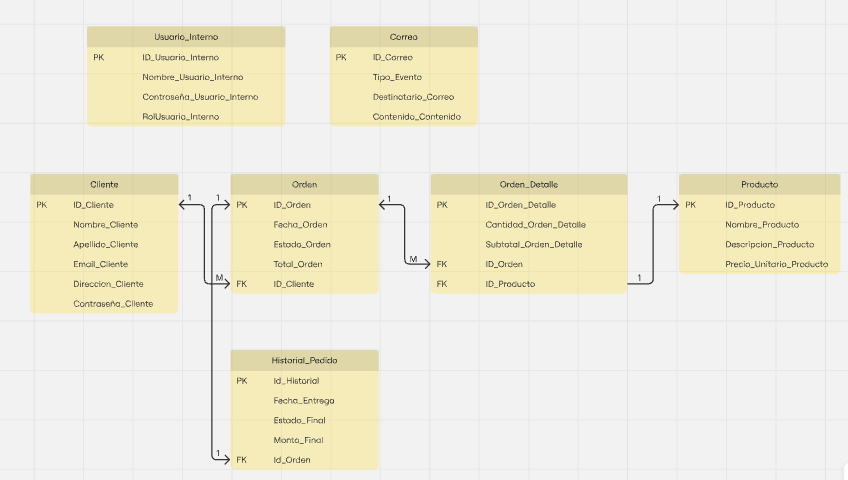


Entrevistador: Matías Antonio Favrat

Entrevistado: Director de Agencia

1. ¿Cuál es el objetivo principal del sitio? (vender, informar, captar leads, otro)
2. ¿Qué hace única a su agencia frente a la competencia?
3. ¿Quién es su cliente ideal? (edad, ubicación, etc)?
4. ¿Qué tipos de paquetes turísticos ofrecerán? (nacionales, internacionales, personalizados, etc.)
5. ¿Planean incluir promociones o descuentos?
6. ¿Quieren permitir que el usuario cree su propio paquete?
7. ¿Qué información debe ver el usuario antes de comprar?
8. ¿Qué tan detallada debe ser la ficha de cada paquete? (itinerario, fotos, mapa, políticas)
9. ¿Desean incluir testimonios o reseñas?
10. ¿Qué páginas debe tener el sitio? (inicio, destinos, contacto, etc.)?
11. ¿Cómo organizarán los paquetes? (por destino, duración, tipo de viaje…)
12. ¿Cuál es el recorrido ideal del usuario desde que entra hasta que compra?
13. ¿Quieren incluir buscador con filtros?
14. ¿Desean que los usuarios puedan registrarse y seguir sus compras?
15. ¿Qué tipo de asistencia se ofrecerá? (chat, FAQs, contacto directo)?
16. ¿Qué estilo visual prefieren? (aventurero, elegante, tropical…)
17. ¿Tienen fotos de calidad o necesitan apoyo con eso?
18. ¿El diseño debe reflejar alguna identidad local, cultural o de marca?
19. ¿El sitio debe ser 100% adaptable a móviles?
20. ¿Cuentan con logotipo, colores y tipografías definidos?
21. ¿Cómo será el proceso de compra o reserva? (en línea o por contacto)
22. ¿Qué métodos de pago desean integrar? (tarjeta, PayPal, transferencia…)
23. ¿Quieren un sistema modular para crear paquetes a medida?
24. ¿Necesitan integrar sistemas de gestión (ventas, clientes, etc.)?
25. ¿Tienen dominio y hosting o necesitan ayuda con eso?
26. ¿Necesitarán soporte técnico post-lanzamiento?
27. ¿Qué nivel de seguridad es obligatorio? (SSL, protección de datos)
28. ¿Hay requisitos legales a reflejar? (políticas, términos, etc.)



* Alcance y requerimientos:
  + Registro, login, logout y recuperación de contraseña.
  + Edición de perfil básico.
  + Catálogo de paquetes turísticos con filtros simples y buscador.
  + Visualización detallada de paquetes.
  + Agregar, eliminar y modificar elementos del carrito (la estrella).
  + Formulario de pago validado + generación de orden de compra.
  + Envío de confirmaciones por correo electrónico (cliente y empresa).
  + Historial de compras por usuario (para que puedan ver qué compraron).
  + Seguridad básica: SSL, protección de datos (no te olvides, que si no te demandan rápido).
  + Responsive design (básico, no un despliegue de portada de revista).
* Diseño mediante Casos de Uso y Diagrama de Entidad-Relación:
  + Casos de Uso Clave:
    - Actores:
      * Usuario (cliente)
      * Sistema (aplicación web)
      * Administrador
    - Casos:
      * Registro de usuario: El usuario crea una cuenta con correo y contraseña. Se valida la información y se almacena.
      * Login / Logout: El usuario inicia sesión para acceder al sistema. Puede cerrar sesión para seguridad.
      * Recuperación de contraseña: El usuario solicita la recuperación mediante correo. Se envía enlace seguro para restablecer contraseña.
      * Navegar catálogo de paquetes: El usuario visualiza paquetes disponibles. Puede usar filtros y buscador para encontrar paquetes.
      * Visualizar detalle de paquete: Mostrar descripción, precio, fechas, servicios incluidos, etc.
      * Agregar paquete al carrito: El usuario añade uno o varios paquetes al carrito.
      * Modificar carrito: El usuario puede eliminar o cambiar cantidad de paquetes en carrito.
      * Realizar pago: El usuario ingresa datos de pago y confirma la compra. Se genera orden y se actualiza el estado.
      * Enviar confirmaciones por correo: Se envía mail al usuario con detalles de compra. Se envía mail al sector interno (ventas/administración).
      * Ver historial de compras: El usuario puede consultar sus compras anteriores
  + DER propuesto (Modificar):
* División de tareas:

| Integrante | Rol Principal | Colaboración Adicional |
| --- | --- | --- |
| Favrat Matías | Gestión del proyecto y Analista Funcional | Apoyo en diseño de base de datos, frontend y documentación técnica. |
| Acosta Lionel | Desarrollo Frontend | Apoyo en testing y documentación de usuario. |
| Kettler Thomas | Desarrollo Backend y Base de Datos | Apoyo en documentación y gestión . |

* Diagrama de tareas a seguir:



## Etapa 2: Implementación Técnica

* Selección e instalación del servidor en Data Center:
  + Acciones:
    - Entorno sobre Ubuntu Server, que es estable e ideal para producción. Configurar dos entornos separados:
      * Servidor de Desarrollo: para probar y hacer QA.
      * Servidor de Producción: seguro, optimizado, solo accesible para despliegue final.
    - Software a instalar:
      * Servidor web: Nginx.
      * Node.js y npm.
      * Certificados SSL con Let’s Encrypt.
      * Configurar puertos seguros.
      * Firewall con UFW o iptables.
* Motor de Base de Datos: PostgreSQL es una excelente elección para este proyecto. Buen rendimiento y muy fiable.
  + Consideraciones:
    - Escalable para muchas transacciones (soporta decenas de miles diarias sin vacilaciones).
    - Crear backups automáticos diarios.
    - Crear los siguientes esquemas iniciales:
    - Usuarios, paquetes, carrito, pagos, mails, etc.
  + Instalación y configuración:
    - Instalar PostgreSQL 15+.
    - Crear un rol específico para la app con privilegios controlados.
    - Habilitar acceso remoto solo desde IPs confiables.
* Tecnologías seleccionadas por capa:

| Capa | Tecnología | Nota |
| --- | --- | --- |
| Frontend | HTML + CSS + BOOTSTRAP | Responsive y rápido para prototipar |
| Backend | Node.js + Express.js | Rápido y flexible |
| Base de Datos | PostgreSQL | Robusto y eficiente |
| Infraestructura | Nginx + Node.js | Combo muy utilizado en las app web |
| Mail | Nodemailer (Node.js) | Para envíos de mails desde el backend |

## 

## Registro del Desarrollo de la Página (Frontend):

1. Organización Inicial:
   * Se creó una carpeta compartida en Google Drive para respaldar y versionar archivos HTML, CSS, documentación y otros recursos.
2. Desarrollo de Interfaz Inicial:
   * Se crearon archivos index.html e index style.css y se guardaron en la carpeta local /frontend.
   * Se aplicó un diseño visual agradable con un degradado entre tonos celeste y azul, brindando una experiencia visual más amigable.
   * Se agregó un título principal y una franja azul para eliminar el aspecto vacío inicial.
3. Adaptabilidad y Navegación:
   * Se implementó un diseño responsive, que adapta la visualización a diferentes dispositivos (móviles, tablets, etc.).
   * Se añadió un menú de navegación central con opciones para iniciar sesión y registrarse.



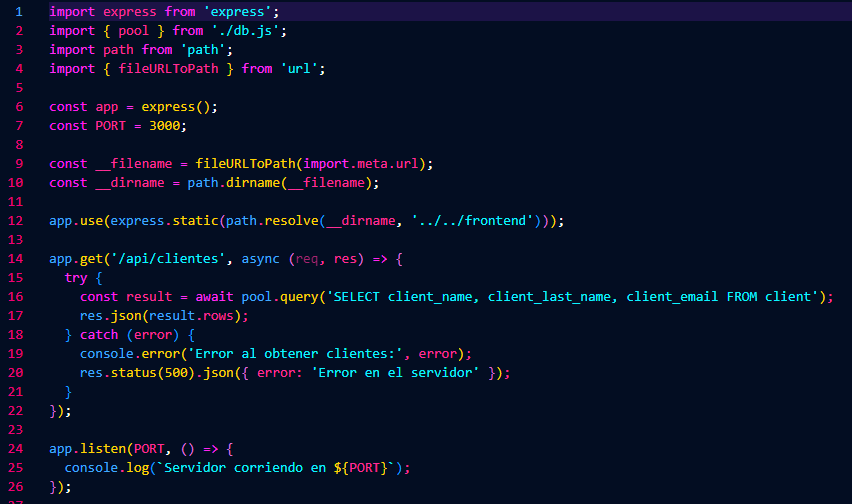
## Desarrollo Backend + Base de Datos:

### Estructura de Carpetas:

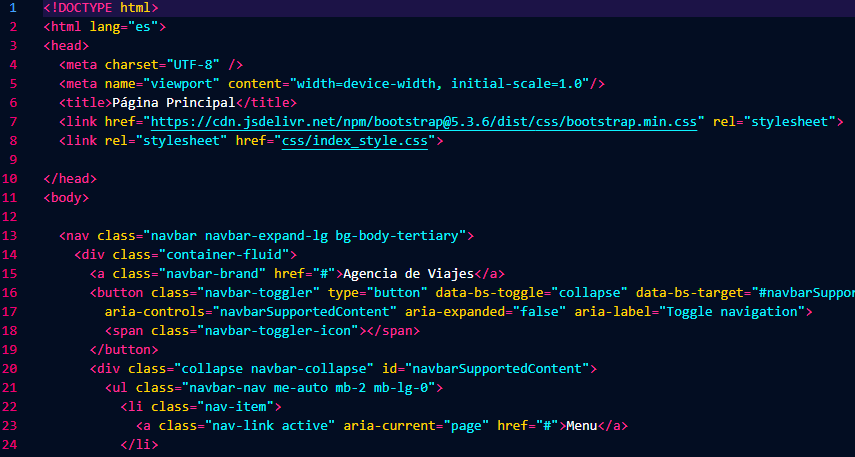
### Conexión a PostgreSQL: Archivo: /backend/src/db.js



### Lógica del Servidor y API REST: Archivo: /backend/src/index.js



### Frontend conectado a API: Archivo: /frontend/index.html





Bibliografía: