Olimpiada de Programación – 2025

E.E.T.P. N°480 Manuel Belgrano

Santa Fe de la Vera Cruz - Santa Fe

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:

Resumen de entrevista simulada al cliente para obtener los requerimientos:

Entrevistador: Matías Antonio Favrat

Entrevistado: Director de Agencia

El sitio web tendrá como objetivo principal captar clientes potenciales interesados en adquirir paquetes turísticos, combinando la venta directa con la oferta de información clara y atractiva. La agencia buscará diferenciarse de la competencia destacando sus valores únicos, ya sea por su enfoque personalizado, experiencia, o propuestas exclusivas. El cliente ideal será definido en función de criterios como edad, ubicación y estilo de viaje, lo cual influirá en el diseño de la oferta.

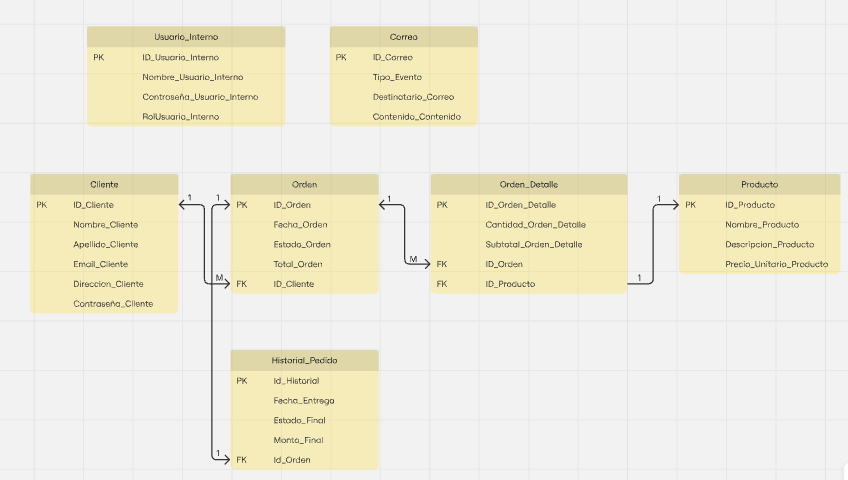
Los paquetes turísticos incluirán opciones nacionales e internacionales, con la posibilidad de personalización según las preferencias del usuario. Se contempla ofrecer promociones y descuentos, así como permitir que el usuario cree su propio paquete a través de un sistema modular. Antes de comprar, el usuario deberá contar con información completa y detallada de cada paquete, incluyendo itinerario, fotos, mapas y políticas de cancelación. Además, se evaluará la inclusión de testimonios o reseñas que refuercen la confianza del visitante.

El sitio estará estructurado en secciones como inicio, destinos, contacto, preguntas frecuentes, y otras según sea necesario. Los paquetes podrán organizarse por destino, duración o tipo de experiencia. Se buscará que el recorrido del usuario sea intuitivo, desde que llega al sitio hasta que realiza una compra, integrando funciones como buscador con filtros, registro de usuarios, y seguimiento de reservas.

En cuanto al soporte, se ofrecerán opciones como chat, contacto directo o secciones de ayuda. El estilo visual reflejará la identidad de la agencia, ya sea elegante, aventurero o tropical, y se evaluará si se cuenta con material fotográfico propio o se necesita apoyo. El diseño será completamente adaptable a móviles y respetará la identidad visual de la marca, incluyendo logotipo, colores y tipografías si ya están definidos.

El proceso de compra podrá realizarse en línea o mediante contacto directo, integrando métodos de pago como tarjeta de crédito, PayPal o transferencia. También se considerará la integración de sistemas de gestión de ventas y clientes. En caso de no contar con dominio o hosting, se ofrecerá ayuda en ese aspecto, así como soporte técnico post-lanzamiento.

Finalmente, el sitio cumplirá con altos estándares de seguridad, incluyendo certificados SSL y protección de datos, además de reflejar todos los requisitos legales como políticas de privacidad, términos y condiciones.

* 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.



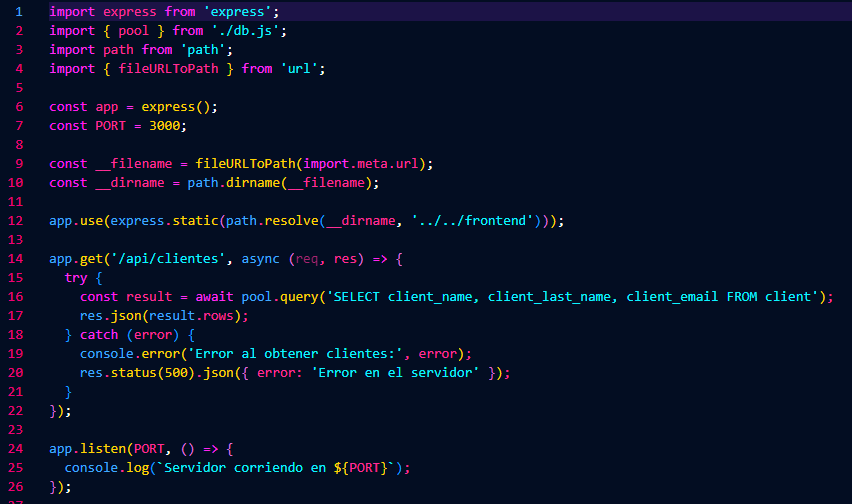
## Desarrollo Backend + Base de Datos:

### Estructura de Carpetas:

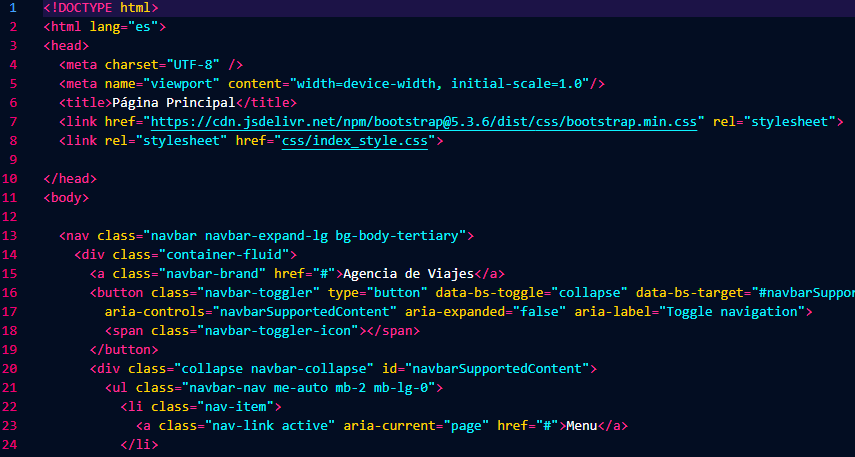
### Conexión a PostgreSQL: Archivo: /backend/src/db.js



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



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





Bibliografía:

Bharti, S. (2024, November 23). Exploring NGINX: a robust web server and reverse proxy. *Medium*. <https://blog.devops.dev/exploring-nginx-a-robust-web-server-and-reverse-proxy-fe1e33ea5749>

Bukhari, S. M. K. (2024, May 28). *A comprehensive guide to NPM (Node Package Manager)*. DEV Community. <https://dev.to/smkbukhari/a-comprehensive-guide-to-npm-node-package-manager-30n4>

Instaclustr. (2025, April 28). *Complete guide to PostgreSQL: Features, use cases, and tutorial*. <https://www.instaclustr.com/education/postgresql/complete-guide-to-postgresql-features-use-cases-and-tutorial/>

SSL Corp. (2023, December 8). *What is SSL/TLS: An In-Depth Guide - SSL.com*. SSL.com. <https://www.ssl.com/article/what-is-ssl-tls-an-in-depth-guide/>

Solanki, M. (2025, May 22). *Standard SSL certificate explained: Domain-Level HTTPS Security*. SSL2BUY. <https://www.ssl2buy.com/wiki/standard-ssl-certificate-guide>

*Ubuntu Server documentation*. (n.d.). Ubuntu Server. <https://documentation.ubuntu.com/server/>

*Index | Node.js v24.2.0 Documentation*. (n.d.). <https://nodejs.org/docs/latest/api/>

*W3Schools.com*. (n.d.). <https://www.w3schools.com/html/html_css.asp>

*W3Schools.com*. (n.d.-b). <https://www.w3schools.com/postgresql/postgresql_intro.php>

*W3Schools.com*. (n.d.-c). <https://www.w3schools.com/js/js_intro.asp>